```
DSECTPM.S (28/02/2015) - DISPLAY DISK SECTORS IN PROTECTED MODE
; ****************************************************************************
; DSECTPM.S (22/02/2015) - DISPLAY DISK SECTORS IN PROTECTED MODE
;                      Retro UNIX 386 v1 - DISK I/O test
; UNIX386.ASM (RETRO UNIX 386 Kernel) - v0.2.0.6 (21/02/2015)
; ----------------------------------------------------------------------------
; NASM version 2.11 (dsectpm.s, unix386.s)
;
; RETRO UNIX 386 (Retro Unix == Turkish Rational Unix)
; Operating System Project (v0.2) by ERDOGAN TAN (Beginning: 24/12/2013)
;
; Derived from 'Retro UNIX 8086 v1' source code by Erdogan Tan
; (v0.1 - Beginning: 11/07/2012)
;
; [ Last Modification: 28/02/2015 ]
;
; Derived from UNIX Operating System (v1.0 for PDP-11)
; (Original) Source Code by Ken Thompson (1971-1972)
; <Bell Laboratories (17/3/1972)>
; <Preliminary Release of UNIX Implementation Document>
;
; Derived from 'UNIX v7/x86' source code by Robert Nordier (1999)
; UNIX V7/x86 source code: see www.nordier.com/v7x86 for details.
;
; ****************************************************************************

; 24/12/2013

; Entering protected mode:
; Derived from 'simple_asm.txt' source code file and
; 'The world of Protected mode' tutorial/article by Gregor Brunmar (2003)
; (gregor.brunmar@home.se)
; http://www.osdever.net/tutorials/view/the-world-of-protected-mode
;


; "The Real, Protected, Long mode assembly tutorial for PCs"
; by Michael Chourdakis (2009)
; http://www.codeproject.com/Articles/45788/
; http://www.michaelchourdakis.com
;

; Global Descriptor Table:
; Derived from 'head.s" source code of Linux v1.0 kernel
; by Linus Torvalds (1991-1992)
;

KLOAD equ 10000h ; Kernel loading address
      ; NOTE: Retro UNIX 8086 v1 /boot code loads kernel at 1000h:0000h
KCODE equ 08h      ; Code segment descriptor (ring 0)
```

```
49                                      KDATA equ 10h      ; Data segment descriptor (ring 0)
50                                      ;UCODE      equ 1Bh ; 18h + 3h  (ring 3)
51                                      ;UDATA      equ 23h ; 20h + 3h  (ring 3)
52                                      ; 27/12/2013
53                                      KEND    equ KLOAD + 65536 ; (28/12/2013) (end of kernel space)
54
55                                      ; IBM PC/AT BIOS ----- 10/06/85 (postequ.inc)
56                                      ;--------- CMOS TABLE LOCATION ADDRESS'S -----------------------------------
57                                      CMOS_SECONDS EQU    00H          ; SECONDS (BCD)
58                                      CMOS_MINUTES EQU    02H          ; MINUTES (BCD)
59                                      CMOS_HOURS   EQU    04H          ; HOURS (BCD)
60                                      CMOS_DAY_WEEK      EQU    06H          ; DAY OF THE WEEK  (BCD)
61                                      CMOS_DAY_MONTH     EQU    07H          ; DAY OF THE MONTH (BCD)
62                                      CMOS_MONTH   EQU    08H          ; MONTH (BCD)
63                                      CMOS_YEAR    EQU    09H          ; YEAR (TWO DIGITS) (BCD)
64                                      CMOS_CENTURY EQU    32H          ; DATE CENTURY BYTE (BCD)
65                                      CMOS_REG_A   EQU    0AH          ; STATUS REGISTER A
66                                      CMOS_REG_D   EQU    0DH          ; STATUS REGISTER D  BATTERY
67                                      ;--------------------------------------
68                                      ;     CMOS EQUATES FOR THIS SYSTEM     ;
69                                      ;-------------------------------------------------------------------------
70                                      CMOS_PORT    EQU    070H         ; I/O ADDRESS OF CMOS ADDRESS PORT
71                                      CMOS_DATA    EQU    071H         ; I/O ADDRESS OF CMOS DATA PORT
72                                      NMI          EQU    10000000B    ; DISABLE NMI INTERRUPTS MASK -
73                                                                       ; HIGH BIT OF CMOS LOCATION ADDRESS
74
75                                      ; Memory Allocation Table Address
76                                      ; 05/11/2014
77                                      ; 31/10/2014
78                                      MEM_ALLOC_TBL      equ    100000h            ; Memory Allocation Table at the end of
79                                                                       ; the 1st 1 MB memory space.
80                                                                       ; (This address must be aligned
81                                                                       ;  on 128 KB boundary, if it will be
82                                                                       ;  changed later.)
83                                                                       ; ((lower 17 bits of 32 bit M.A.T.
84                                                                       ;   address must be ZERO)).
85                                                                       ; ((((Reason: 32 bit allocation
86                                                                       ;     instructions, dword steps)))
87                                                                       ; (((byte >> 12 --> page >> 5)))
88                                      ;04/11/2014
89                                      PDE_A_PRESENT      equ    1            ; Present flag for PDE
90                                      PDE_A_WRITABLE     equ    2            ; Writable (write permission) flag
91                                      PDE_A_USER   equ    4            ; User (non-system/kernel) page flag
92                                      ;
93                                      PTE_A_PRESENT      equ    1            ; Present flag for PTE
94                                      PTE_A_WRITABLE     equ    2            ; Writable (write permission) flag
95                                      PTE_A_USER   equ    4            ; User (non-system/kernel) page flag
96
97
```

```
 98                                     ; 17/02/2015 (unix386.s)
 99                                     ; 10/12/2014 - 30/12/2014 (0B000h -> 9000h) (dsectrm2.s)
100                                     DPT_SEGM equ 09000h  ; FDPT segment (EDD v1.1, EDD v3)
101                                     ;
102                                     HD0_DPT      equ 0    ; Disk parameter table address for hd0
103                                     HD1_DPT      equ 32          ; Disk parameter table address for hd1
104                                     HD2_DPT      equ 64          ; Disk parameter table address for hd2
105                                     HD3_DPT      equ 96          ; Disk parameter table address for hd3
106                                     
107                                     
108                                     ; FDPT (Phoenix, Enhanced Disk Drive Specification v1.1, v3.0)
109                                     ;      (HDPT: Programmer's Guide to the AMIBIOS, 1993)
110                                     ;
111                                     FDPT_CYLS   equ 0 ; 1 word, number of cylinders
112                                     FDPT_HDS    equ 2 ; 1 byte, number of heads
113                                     FDPT_TT           equ 3 ; 1 byte, A0h = translated FDPT with logical values
114                                                       ; otherwise it is standard FDPT with physical values
115                                     FDPT_PCMP   equ 5 ; 1 word, starting write precompensation cylinder
116                                                       ; (obsolete for IDE/ATA drives)
117                                     FDPT_CB           equ 8 ; 1 byte, drive control byte
118                                                       ; Bits 7-6 : Enable or disable retries (00h = enable)
119                                                       ; Bit 5      : 1 = Defect map is located at last cyl. + 1
120                                                       ; Bit 4 : Reserved. Always 0
121                                                       ; Bit 3 : Set to 1 if more than 8 heads
122                                                       ; Bit 2-0 : Reserved. Alsways 0
123                                     FDPT_LZ           equ 12 ; 1 word, landing zone (obsolete for IDE/ATA drives)
124                                     FDPT_SPT    equ 14 ; 1 byte, sectors per track
125                                     
126                                     ; Floppy Drive Parameters Table (Programmer's Guide to the AMIBIOS, 1993)
127                                     ; (11 bytes long) will be used by diskette handler/bios
128                                     ; which is derived from IBM PC-AT BIOS (DISKETTE.ASM, 21/04/1986).

129                                     
130                                     
131                                     
132                                     [BITS 16]       ; We need 16-bit intructions for Real mode
133                                     
134                                     [ORG 0]
135                                           ; 12/11/2014
136                                           ; Save boot drive number (that is default root drive)
137 00000000 8816[1249]                       mov    [boot_drv], dl ; physical drv number
138                                     
139                                           ; Determine installed memory
140                                           ; 31/10/2014
141                                           ;
142 00000004 B801E8                           mov    ax, 0E801h ; Get memory size
143 00000007 CD15                             int    15h       ; for large configurations
144 00000009 7308                             jnc    short chk_ms
145 0000000B B488                             mov    ah, 88h    ; Get extended memory size
```

```
146 0000000D CD15                       int   15h
147                                     ;
148                                     ;mov  al, 17h      ; Extended memory (1K blocks) low byte
149                                     ;out  70h, al ; select CMOS register
150                                     ;in   al, 71h ; read data (1 byte)
151                                     ;mov  cl, al
152                                     ;mov  al, 18h ; Extended memory (1K blocks) high byte
153                                     ;out  70h, al ; select CMOS register
154                                     ;in   al, 71h ; read data (1 byte)
155                                     ;mov  ch, al
156                                     ;
157 0000000F 89C1                       mov   cx, ax
158 00000011 31D2                       xor   dx, dx
159                                   chk_ms:
160 00000013 890E[1C49]                 mov   [mem_1m_1k], cx
161 00000017 8916[1E49]                 mov   [mem_16m_64k], dx
162                                     ; 05/11/2014
163                                     ;and  dx, dx
164                                     ;jz   short L2
165 0000001B 81F90004                     cmp    cx, 1024
166 0000001F 7351                       jnb   short L0
167                                            ; insufficient memory_error
168                                            ; Minimum 2 MB memory is needed...
169                                     ; 05/11/2014
170                                     ; (real mode error printing)
171 00000021 FB                         sti
172 00000022 BE[3600]                   mov   si, msg_out_of_memory
173 00000025 BB0700                     mov   bx, 7
174 00000028 B40E                       mov   ah, 0Eh      ; write tty
175                                   oom_1:
176 0000002A AC                         lodsb
177 0000002B 08C0                       or    al, al
178 0000002D 7404                       jz    short oom_2
179 0000002F CD10                       int   10h
180 00000031 EBF7                       jmp   short oom_1
181                                   oom_2:
182 00000033 F4                           hlt
183 00000034 EBFD                       jmp   short oom_2
184
185                                   ; 05/11/2014
186                                   msg_out_of_memory:
187 00000036 070D0A                     db    07h, 0Dh, 0Ah
188 00000039 496E73756666696369-          db      'Insufficient memory ! (Minimum 2 MB memory is needed.)'
189 00000042 656E74206D656D6F72-
190 0000004B 79202120284D696E69-
191 00000054 6D756D2032204D4220-
192 0000005D 6D656D6F7279206973-
193 00000066 206E65656465642E29
194 0000006F 0D0A00                     db    0Dh, 0Ah, 0
```

```
                                        ;
                                ; DISK I/O SYSTEM INITIALIZATION - Erdogan Tan (Retro UNIX 386 v1 project)

                                ; ///////// DISK I/O SYSTEM STRUCTURE INITIALIZATION ///////////////

                                        ; 10/12/2014 - 02/02/2015 - dsectrm2.s
                                L0:
                                        ; 22/02/2015 - dsectpm.s
00000072 BE[3248]                       mov     si, prg_msg
00000075 BB0700                         mov     bx, 7
00000078 B40E                           mov     ah, 0eh
                                startmsgl:
0000007A AC                             lodsb
0000007B 20C0                           and     al, al
0000007D 7404                           jz      short startmsg_ok
0000007F CD10                           int     10h
00000081 EBF7                           jmp     short startmsgl
                                startmsg_ok:
                                        ;
                                        ; 12/11/2014 (Retro UNIX 386 v1 - beginning)
                                        ; Detecting disk drives... (by help of ROM-BIOS)
00000083 BA7F00                         mov     dx, 7Fh
                                L1:
00000086 FEC2                           inc     dl
00000088 B441                           mov     ah, 41h ; Check extensions present
                                                        ; Phoenix EDD v1.1 - EDD v3
0000008A BBAA55                         mov     bx, 55AAh
0000008D CD13                           int     13h
0000008F 721A                           jc      short L2
00000091 81FB55AA                       cmp     bx, 0AA55h
00000095 7514                           jne     short L2
00000097 FE06[1449]                     inc     byte [hdc]   ; count of hard disks (EDD present)
0000009B 8816[3048]                       mov     [last_drv], dl  ; last hard disk number
0000009F BB[9848]                       mov     bx, hd0_type - 80h
000000A2 01D3                           add     bx, dx
000000A4 880F                           mov     [bx], cl ; Interface support bit map in CX
                                                         ; Bit 0 - 1, Fixed disk access subset ready
                                                         ; Bit 1 - 1, Drv locking and ejecting ready
                                                         ; Bit 2 - 1, Enhanced Disk Drive Support
                                                              ;           (EDD) ready (DPTE ready)
                                                         ; Bit 3 - 1, 64bit extensions are present
                                                              ;           (EDD-3)
                                                         ; Bit 4 to 15 - 0, Reserved
000000A6 80FA83                         cmp     dl, 83h       ; drive number < 83h
000000A9 72DB                           jb      short L1
                                L2:
                                        ; 23/11/2014
                                        ; 19/11/2014
000000AB 30D2                           xor     dl, dl  ; 0
```

```
000000AD 66BE[16490000]                 mov   esi, fd0_type
                                    L3:
                                        ; 14/01/2015
000000B3 8816[1349]                     mov   [drv], dl
                                        ;
000000B7 B408                           mov   ah, 08h ; Return drive parameters
000000B9 CD13                           int   13h
000000BB 7215                           jc    short L4
                                              ; BL = drive type (for floppy drives)
                                              ; DL = number of floppy drives
                                              ;
                                              ; ES:DI = Address of DPT from BIOS
                                              ;
000000BD 67881E                         mov   [esi], bl ;  Drive type
                                                     ; 4 = 1.44 MB, 80 track, 3 1/2"
                                        ; 14/01/2015
000000C0 E8BD02                         call  set_disk_parms
                                        ; 10/12/2014
000000C3 6681FE[16490000]               cmp   esi, fd0_type
000000CA 7706                           ja    short L4
000000CC 6646                           inc   esi ; fd1_type
000000CE B201                           mov   dl, 1
000000D0 EBE1                           jmp   short L3
                                    L4:
                                        ; Older BIOS (INT 13h, AH = 48h is not available)
000000D2 B27F                           mov   dl, 7Fh
                                        ; 24/12/2014 (Temporary)
000000D4 803E[1449]00                   cmp   byte [hdc], 0 ; EDD present or not ?
000000D9 0F879000                         ja      L10       ; yes, all fixed disk operations
                                                  ; will be performed according to
                                                  ; present EDD specification
                                    L6:
000000DD FEC2                           inc   dl
000000DF 8816[1349]                       mov     [drv], dl
000000E3 8816[3048]                       mov     [last_drv], dl ; 14/01/2015
000000E7 B408                           mov   ah, 08h ; Return drive parameters
000000E9 CD13                           int   13h    ; (conventional function)
000000EB 0F829401                         jc      L13       ; fixed disk drive not ready
000000EF 8816[1449]                       mov     [hdc], dl ; number of drives
                                        ;; 14/01/2013
                                        ;;push cx
000000F3 E88A02                         call  set_disk_parms
                                        ;;pop  cx
                                        ;
                                        ;;and  cl, 3Fh       ; sectors per track (bits 0-6)
000000F6 8A16[1349]                       mov     dl, [drv]
000000FA BB0401                         mov   bx, 65*4 ; hd0 parameters table (INT 41h)
000000FD 80FA80                         cmp   dl, 80h
00000100 7603                           jna   short L7
```

```
293 00000102 83C314                     add   bx, 5*4       ; hd1 parameters table (INT 46h)
294                                 L7:
295 00000105 31C0                       xor   ax, ax
296 00000107 8ED8                       mov   ds, ax
297 00000109 8B37                         mov   si, [bx]
298 0000010B 8B4702                       mov   ax, [bx+2]
299 0000010E 8ED8                       mov   ds, ax
300 00000110 3A4C0E                       cmp   cl, [si+FDPT_SPT] ; sectors per track
301 00000113 0F856801                     jne   L12 ; invalid FDPT
302 00000117 BF0000                     mov   di, HD0_DPT
303 0000011A 80FA80                     cmp   dl, 80h
304 0000011D 7603                       jna   short L8
305 0000011F BF2000                     mov   di, HD1_DPT
306                                 L8:
307                                     ; 30/12/2014
308 00000122 B80090                     mov   ax, DPT_SEGM
309 00000125 8EC0                       mov   es, ax
310                                     ; 24/12/2014
311 00000127 B90800                     mov   cx, 8
312 0000012A F3A5                       rep   movsw  ; copy 16 bytes to the kernel's DPT location
313 0000012C 8CC8                       mov   ax, cs
314 0000012E 8ED8                       mov   ds, ax
315                                     ; 02/02/2015
316 00000130 8A0E[1349]                   mov   cl, [drv]
317 00000134 88CB                       mov   bl, cl
318 00000136 B8F001                     mov   ax, 1F0h
319 00000139 80E301                     and   bl, 1
320 0000013C 7406                       jz    short L9
321 0000013E C0E304                     shl   bl, 4
322 00000141 2D8000                     sub   ax, 1F0h-170h
323                                 L9:
324 00000144 AB                         stosw  ; I/O PORT Base Address (1F0h, 170h)
325 00000145 050602                     add   ax, 206h
326 00000148 AB                         stosw  ; CONTROL PORT Address (3F6h, 376h)
327 00000149 88D8                       mov   al, bl
328 0000014B 04A0                       add   al, 0A0h
329 0000014D AA                         stosb  ; Device/Head Register upper nibble
330                                     ;
331 0000014E FE06[1349]                 inc   byte [drv]
332 00000152 BB[9848]                   mov   bx, hd0_type - 80h
333 00000155 01CB                       add   bx, cx
334 00000157 800F80                       or    byte [bx], 80h  ; present sign (when lower nibble is 0)
335 0000015A A0[1449]                   mov   al, [hdc]
336 0000015D FEC8                       dec   al
337 0000015F 0F842001                     jz    L13
338 00000163 80FA80                     cmp   dl, 80h
339 00000166 0F8673FF                     jna   L6
340 0000016A E91601                       jmp   L13
341                                 L10:
```

```
342 0000016D FEC2                       inc    dl
343                                     ; 25/12/2014
344 0000016F 8816[1349]                 mov    [drv], dl
345 00000173 B408                       mov    ah, 08h ; Return drive parameters
346 00000175 CD13                       int    13h    ; (conventional function)
347 00000177 0F820801                     jc      L13
348                                     ; 14/01/2015
349 0000017B 8A16[1349]                 mov    dl, [drv]
350 0000017F 52                         push   dx
351 00000180 51                         push   cx
352 00000181 E8FC01                     call   set_disk_parms
353 00000184 59                         pop    cx
354 00000185 5A                         pop    dx
355                                     ;
356 00000186 66BE[D04A0000]             mov    esi, _end ; 30 byte temporary buffer address
357                                                      ; at the '_end' of kernel.
358 0000018C 67C7061E00                 mov    word [esi], 30
359 00000191 B448                       mov    ah, 48h       ; Get drive parameters (EDD function)
360 00000193 CD13                       int    13h
361 00000195 0F82EA00                     jc      L13
362                                     ; 14/01/2015
363 00000199 6629DB                     sub    ebx, ebx
364 0000019C 88D3                       mov    bl, dl
365 0000019E 80EB80                     sub    bl, 80h
366 000001A1 6681C3[18490000]           add    ebx, hd0_type
367 000001A8 678A03                     mov    al, [ebx]
368 000001AB 0C80                       or     al, 80h
369 000001AD 678803                     mov    [ebx], al
370 000001B0 6681EB[16490000]           sub    ebx, hd0_type - 2 ; 15/01/2015
371 000001B7 6681C3[24480000]           add    ebx, drv_status
372 000001BE 678803                     mov    [ebx], al
373 000001C1 66678B4610                 mov    eax, [esi+16]
374                                     ; 28/02/2015
375 000001C6 6621C0                     and    eax, eax
376 000001C9 7416                       jz     short L10_A0h
377                                                ; 'CHS only' disks on EDD system
378                                                ;  are reported with ZERO disk size
379 000001CB 6681EB[24480000]           sub    ebx, drv_status
380 000001D2 66C1E302                   shl    ebx, 2
381 000001D6 6681C3[EA480000]           add    ebx, disk_size
382 000001DD 66678903                   mov    [ebx], eax
383                                   L10_A0h: ; Jump here to fix a ZERO (LBA) disk size problem
384                                          ; for CHS disks (28/02/2015)
385                                     ; 30/12/2014
386 000001E1 BF0000                     mov    di, HD0_DPT
387 000001E4 88D0                       mov    al, dl
388 000001E6 83E003                     and    ax, 3
389 000001E9 C0E005                     shl    al, 5 ; *32
390 000001EC 01C7                       add    di, ax
```

```
391 000001EE B80090                    mov   ax, DPT_SEGM
392 000001F1 8EC0                      mov   es, ax
393                                    ;
394 000001F3 88E8                      mov   al, ch ; max. cylinder number (bits 0-7)
395 000001F5 88CC                      mov   ah, cl
396 000001F7 C0EC06                    shr   ah, 6  ; max. cylinder number (bits 8-9)
397 000001FA 40                        inc   ax     ; logical cylinders (limit 1024)
398 000001FB AB                        stosw
399 000001FC 88F0                      mov   al, dh ; max. head number
400 000001FE FEC0                      inc   al
401 00000200 AA                        stosb        ; logical heads (limits 256)
402 00000201 B0A0                      mov   al, 0A0h ; Indicates translated table
403 00000203 AA                        stosb
404 00000204 8A440C                    mov   al, [si+12]
405 00000207 AA                        stosb        ; physical sectors per track
406 00000208 31C0                      xor   ax, ax
407                                    ;dec  ax     ; 02/01/2015
408 0000020A AB                        stosw        ; precompensation (obsolete)
409                                    ;xor  al, al ; 02/01/2015
410 0000020B AA                        stosb        ; reserved
411 0000020C B008                      mov   al, 8  ; drive control byte
412                                                   ; (do not disable retries,
413                                                  ; more than 8 heads)
414 0000020E AA                        stosb
415 0000020F 8B4404                    mov   ax, [si+4]
416 00000212 AB                        stosw        ; physical number of cylinders
417                                    ;push ax     ; 02/01/2015
418 00000213 8A4408                    mov   al, [si+8]
419 00000216 AA                        stosb        ; physical num. of heads (limit 16)
420 00000217 29C0                      sub   ax, ax
421                                    ;pop  ax     ; 02/01/2015
422 00000219 AB                        stosw        ; landing zone (obsolete)
423 0000021A 88C8                      mov   al, cl ; logical sectors per track (limit 63)
424 0000021C 243F                      and   al, 3Fh
425 0000021E AA                        stosb
426                                    ;sub  al, al ; checksum
427                                    ;stosb
428                                    ;
429 0000021F 83C61A                    add   si, 26   ; (BIOS) DPTE address pointer
430 00000222 AD                        lodsw
431 00000223 50                        push  ax     ; (BIOS) DPTE offset
432 00000224 AD                        lodsw
433 00000225 50                        push  ax     ; (BIOS) DPTE segment
434                                    ;
435                                    ; checksum calculation
436 00000226 89FE                      mov   si, di
437 00000228 06                        push  es
438 00000229 1F                        pop   ds
439                                    ;mov  cx, 16
```

```
0000022A B90F00                         mov    cx, 15
0000022D 29CE                           sub    si, cx
0000022F 30E4                           xor    ah, ah
                                        ;del   cl
                                L11:
00000231 AC                             lodsb
00000232 00C4                           add    ah, al
00000234 E2FB                           loop   L11
                                        ;
00000236 88E0                           mov    al, ah
00000238 F6D8                           neg    al     ; -x+x = 0
0000023A AA                             stosb         ; put checksum in byte 15 of the tbl
                                        ;
0000023B 1F                             pop    ds     ; (BIOS) DPTE segment
0000023C 5E                             pop    si     ; (BIOS) DPTE offset
                                        ;
                                        ; 23/02/2015
0000023D 57                             push   di
                                        ; ES:DI points to DPTE (FDPTE) location
                                        ;mov   cx, 8
0000023E B108                           mov    cl, 8
00000240 F3A5                           rep    movsw
                                        ;
                                        ; 23/02/2015
                                        ; (P)ATA drive and LBA validation
                                        ; (invalidating SATA drives and setting
                                        ; CHS type I/O for old type fixed disks)
00000242 5B                             pop    bx
00000243 8CC8                           mov    ax, cs
00000245 8ED8                           mov    ds, ax
00000247 268B07                         mov    ax, [es:bx]
0000024A 3DF001                         cmp    ax, 1F0h
0000024D 7413                           je     short L11a
0000024F 3D7001                         cmp    ax, 170h
00000252 740E                           je     short L11a
                                        ; invalidation
                                        ; (because base port addres is not 1F0h or 170h)
00000254 30FF                           xor    bh, bh
00000256 88D3                           mov    bl, dl
00000258 80EB80                         sub    bl, 80h
0000025B C687[1849]00                     mov     byte [bx+hd0_type], 0 ; not a valid disk drive !
                                        ;or    [bx+drv_status+2], 0F0h ; (failure sign)
00000260 EB14                           jmp    short L11b
                                L11a:
                                        ; LBA validation
00000262 268A4704                       mov    al, [es:bx+4] ; Head register upper nibble
00000266 A840                           test   al, 40h ; LBA bit (bit 6)
00000268 750C                           jnz    short L11b ; LBA type I/O is OK! (E0h or F0h)
                                        ; force CHS type I/O for this drive (A0h or B0h)
```

```
489 0000026A 28FF                       sub    bh, bh
490 0000026C 88D3                       mov    bl, dl
491 0000026E 80EB80                     sub    bl, 80h ; 26/02/2015
492 00000271 80A7[2648]FE               and    byte [bx+drv_status+2], 0FEh ; clear bit 0
493                                                     ; bit 0 = LBA ready bit
494                                     ; 'read_disk_sector' procedure will check this bit !
495                                L11b:
496 00000276 3A16[3048]                 cmp    dl, [last_drv] ; 25/12/2014
497 0000027A 7307                         jnb     short L13
498 0000027C E9EEFE                       jmp     L10
499                                L12:
500                                     ; Restore data registers
501 0000027F 8CC8                       mov    ax, cs
502 00000281 8ED8                       mov    ds, ax
503                                L13:
504                                     ; 13/12/2014
505 00000283 0E                         push   cs
506 00000284 07                         pop    es
507                                L14:
508 00000285 B411                       mov    ah, 11h
509 00000287 CD16                       int    16h
510 00000289 7406                       jz     short L15
511 0000028B B010                       mov    al, 10h
512 0000028D CD16                       int    16h
513 0000028F EBF4                       jmp    short L14
514                                L15:
515                                ; //////
516                                     ; 24/11/2014
517                                     ; 19/11/2014
518                                     ; 14/11/2014
519                                     ; Temporary code for disk searching code check
520                                     ;
521                                     ; This code will show existing (usable) drives and also
522                                     ; will show EDD interface support status for hard disks
523                                     ; (If status bit 7 is 1, Identify Device info is ready,
524                                     ; no need to get it again in protected mode...)
525                                     ;
526                                     ; 13/11/2014
527 00000291 BB0700                     mov    bx, 7
528 00000294 B40E                       mov    ah, 0Eh
529 00000296 A0[1649]                   mov    al, [fd0_type]
530 00000299 20C0                       and    al, al
531 0000029B 743D                       jz     short L15a
532 0000029D 88C2                       mov    dl, al
533 0000029F B046                       mov    al, 'F'
534 000002A1 CD10                       int    10h
535 000002A3 B044                       mov    al, 'D'
536 000002A5 CD10                       int    10h
537 000002A7 B030                       mov    al, '0'
```

```
538 000002A9 CD10                       int   10h
539 000002AB B020                       mov   al, ' '
540 000002AD CD10                       int   10h
541 000002AF E8B700                     call  L15c
542 000002B2 B020                       mov   al, ' '
543 000002B4 CD10                       int   10h
544                                     ;
545 000002B6 A0[1749]                   mov   al, [fd1_type]
546 000002B9 20C0                       and   al, al
547 000002BB 741D                       jz    short L15a
548 000002BD 88C2                       mov   dl, al
549 000002BF B046                       mov   al, 'F'
550 000002C1 CD10                       int   10h
551 000002C3 B044                       mov   al, 'D'
552 000002C5 CD10                       int   10h
553 000002C7 B031                       mov   al, '1'
554 000002C9 CD10                       int   10h
555 000002CB B020                       mov   al, ' '
556 000002CD CD10                       int   10h
557 000002CF E89700                     call  L15c
558 000002D2 B020                       mov   al, ' '
559 000002D4 CD10                       int   10h
560 000002D6 B020                       mov   al, ' '
561 000002D8 CD10                       int   10h
562                                 L15a:
563 000002DA A0[1849]                   mov   al, [hd0_type]
564 000002DD 20C0                       and   al, al
565 000002DF 7479                       jz    short L15b
566 000002E1 88C2                       mov   dl, al
567 000002E3 B048                       mov   al, 'H'
568 000002E5 CD10                       int   10h
569 000002E7 B044                       mov   al, 'D'
570 000002E9 CD10                       int   10h
571 000002EB B030                       mov   al, '0'
572 000002ED CD10                       int   10h
573 000002EF B020                       mov   al, ' '
574 000002F1 CD10                       int   10h
575 000002F3 E87300                     call  L15c
576 000002F6 B020                       mov   al, ' '
577 000002F8 CD10                       int   10h
578                                     ;
579 000002FA A0[1949]                   mov   al, [hd1_type]
580 000002FD 20C0                       and   al, al
581 000002FF 7459                       jz    short L15b
582 00000301 88C2                       mov   dl, al
583 00000303 B048                       mov   al, 'H'
584 00000305 CD10                       int   10h
585 00000307 B044                       mov   al, 'D'
586 00000309 CD10                       int   10h
```

```
587 0000030B B031                       mov   al, '1'
588 0000030D CD10                       int   10h
589 0000030F B020                       mov   al, ' '
590 00000311 CD10                       int   10h
591 00000313 E85300                     call  L15c
592 00000316 B020                       mov   al, ' '
593 00000318 CD10                       int   10h
594                                     ;
595 0000031A A0[1A49]                   mov   al, [hd2_type]
596 0000031D 20C0                       and   al, al
597 0000031F 7439                       jz    short L15b
598 00000321 88C2                       mov   dl, al
599 00000323 B048                       mov   al, 'H'
600 00000325 CD10                       int   10h
601 00000327 B044                       mov   al, 'D'
602 00000329 CD10                       int   10h
603 0000032B B032                       mov   al, '2'
604 0000032D CD10                       int   10h
605 0000032F B020                       mov   al, ' '
606 00000331 CD10                       int   10h
607 00000333 E83300                     call  L15c
608 00000336 B020                       mov   al, ' '
609 00000338 CD10                       int   10h
610                                     ;
611 0000033A A0[1B49]                   mov   al, [hd3_type]
612 0000033D 20C0                       and   al, al
613 0000033F 7419                       jz    short L15b
614 00000341 88C2                       mov   dl, al
615 00000343 B048                       mov   al, 'H'
616 00000345 CD10                       int   10h
617 00000347 B044                       mov   al, 'D'
618 00000349 CD10                       int   10h
619 0000034B B033                       mov   al, '3'
620 0000034D CD10                       int   10h
621 0000034F B020                       mov   al, ' '
622 00000351 CD10                       int   10h
623 00000353 E81300                     call  L15c
624 00000356 B020                       mov   al, ' '
625 00000358 CD10                       int   10h
626                                     ;
627                                 L15b:
628 0000035A B00D                       mov   al, 0Dh
629 0000035C CD10                       int   10h
630 0000035E B00A                       mov   al, 0Ah
631 00000360 CD10                       int   10h
632                                     ;
633 00000362 30E4                       xor   ah, ah
634 00000364 CD16                       int   16h
635                                     ;
```

```
636                                             ;jmp     short L16
637 00000366 E99A00                             jmp     L16
638                                             ;
639                                     L15c:
640 00000369 88D6                               mov     dh, dl
641 0000036B C0EE04                             shr     dh, 4
642 0000036E 80C630                             add     dh, 30h
643 00000371 80E20F                             and     dl, 15
644 00000374 80C230                             add     dl, 30h
645 00000377 88F0                               mov     al, dh
646 00000379 CD10                               int     10h
647 0000037B 88D0                               mov     al, dl
648 0000037D CD10                               int     10h
649 0000037F C3                                 retn
650                                             ;
651                                             ; end of temporary code for disk searching code check
652
653                                     ; //////
654
655                                     set_disk_parms:
656                                             ; 14/01/2015
657                                             ;push  ebx
658 00000380 6629DB                             sub     ebx, ebx
659 00000383 8A1E[1349]                         mov     bl, [drv]
660 00000387 80FB80                             cmp     bl, 80h
661 0000038A 7203                               jb      short sdp0
662 0000038C 80EB7E                             sub     bl, 7Eh
663                                     sdp0:
664 0000038F 6681C3[24480000]                   add     ebx, drv_status
665 00000396 67C60380                           mov     byte [ebx], 80h ; 'Present' flag
666                                             ;
667 0000039A 88E8                               mov     al, ch ; last cylinder (bits 0-7)
668 0000039C 88CC                               mov     ah, cl ;
669 0000039E C0EC06                             shr     ah, 6  ; last cylinder (bits 8-9)
670 000003A1 6681EB[24480000]                   sub     ebx, drv_status
671 000003A8 D0E3                               shl     bl, 1
672 000003AA 6681C3[C6480000]                   add     ebx, cylinders
673 000003B1 40                                 inc     ax  ; convert max. cyl number to cyl count
674 000003B2 678903                             mov     [ebx], ax
675 000003B5 50                                 push    ax ; ** cylinders
676 000003B6 6681EB[C6480000]                   sub     ebx, cylinders
677 000003BD 6681C3[D2480000]                   add     ebx, heads
678 000003C4 30E4                               xor     ah, ah
679 000003C6 88F0                               mov     al, dh ; heads
680 000003C8 40                                 inc     ax
681 000003C9 678903                             mov     [ebx], ax
682 000003CC 6681EB[D2480000]                   sub     ebx, heads
683 000003D3 6681C3[DE480000]                   add     ebx, spt
684 000003DA 30ED                               xor     ch, ch
```

```
685 000003DC 80E13F                     and    cl, 3Fh      ; sectors (bits 0-6)
686 000003DF 67890B                     mov    [ebx], cx
687 000003E2 6681EB[DE480000]           sub    ebx, spt
688 000003E9 66D1E3                     shl    ebx, 1
689 000003EC 6681C3[EA480000]           add    ebx, disk_size
690                                     ; LBA size = cylinders * heads * secpertrack
691 000003F3 F7E1                       mul    cx
692 000003F5 89C2                       mov    dx, ax ; heads*spt
693 000003F7 58                         pop    ax ; ** cylinders
694 000003F8 48                         dec    ax ; 1 cylinder reserved (!?)
695 000003F9 F7E2                       mul    dx ; cylinders * (heads*spt)
696 000003FB 678903                     mov    [ebx], ax
697 000003FE 67895302                   mov    [ebx+2], dx
698                                     ;
699                                     ;pop   ebx
700 00000402 C3                         retn
701
702                                 ; End Of DISK I/O SYSTEM STRUCTURE INITIALIZATION /// 06/02/2015
703
704                                 L16:
705                                 _L:    ; 12/02/2015 (unix386.s)
706                                        ; ----------------------------------------------------
707                                        ; 28/11/2014 (dsectrm2.s)
708 00000403 30FF                       xor    bh, bh
709 00000405 B403                       mov    ah, 03h      ; get cursor position and shape
710 00000407 CD10                       int    10h
711                                     ;mov   [cursor_posb], dx ; position ;; 16/02/2015
712 00000409 890E[2248]                 mov    [cursor_shp], cx ; shape
713                                     ; ----------------------------------------------------
714                                     ;
715                                     ; 10/11/2014
716 0000040D FA                         cli    ; Disable interrupts (clear interrupt flag)
717                                            ; Reset Interrupt MASK Registers (Master&Slave)
718                                     ;mov   al, 0FFh     ; mask off all interrupts
719                                     ;out   21h, al             ; on master PIC (8259)
720                                     ; jmp  $+2  ; (delay)
721                                     ;out   0A1h, al     ; on slave PIC (8259)
722                                     ;
723                                     ; Disable NMI
724 0000040E B080                       mov    al, 80h
725 00000410 E670                       out    70h, al             ; set bit 7 to 1 for disabling NMI
726                                      ;23/02/2015
727 00000412 90                         nop
728 00000413 E471                       in     al, 71h             ; read in 71h just after writing out to 70h
729                                                                ; for preventing unknown state (!?)
730                                     ; 20/08/2014
731                                     ; Moving the kernel 64 KB back (to physical address 0)
732                                     ; DS = CS = 1000h
733                                     ; 05/11/2014
```

```
734 00000415 31C0                       xor    ax, ax
735 00000417 8EC0                       mov    es, ax ; ES = 0
736                                     ;
737 00000419 B90040                     mov    cx, (KEND - KLOAD)/4
738 0000041C 31F6                       xor    si, si
739 0000041E 31FF                       xor    di, di
740 00000420 F366A5                     rep    movsd
741                                     ;
742 00000423 06                         push   es ; 0
743 00000424 68[2804]                   push   L17
744 00000427 CB                         retf
745                                     ;
746                                 L17:
747                                     ; Turn off the floppy drive motor
748 00000428 BAF203                       mov     dx, 3F2h
749 0000042B EE                           out     dx, al ; 0 ; 31/12/2013
750
751                                     ; Enable access to memory above one megabyte
752                                 L18:
753 0000042C E464                       in     al, 64h
754 0000042E A802                       test   al, 2
755 00000430 75FA                         jnz     short L18
756 00000432 B0D1                       mov    al, 0D1h     ; Write output port
757 00000434 E664                       out    64h, al
758                                 L19:
759 00000436 E464                       in     al, 64h
760 00000438 A802                       test   al, 2
761 0000043A 75FA                         jnz     short L19
762 0000043C B0DF                       mov    al, 0DFh     ; Enable A20 line
763 0000043E E660                       out    60h, al
764                                 ;L20:
765                                     ;
766                                     ; Load global descriptor table register
767
768                                       ;mov     ax, cs
769                                       ;mov     ds, ax
770
771 00000440 2E0F0116[300E]               lgdt    [cs:gdtd]
772
773 00000446 0F20C0                       mov     eax, cr0
774                                     ; or   eax, 1
775 00000449 40                         inc     ax
776 0000044A 0F22C0                     mov     cr0, eax
777
778                                     ; Jump to 32 bit code
779
780 0000044D 66                         db 66h                  ; Prefix for 32-bit
781 0000044E EA                         db 0EAh             ; Opcode for far jump
782 0000044F [55040000]                 dd StartPM          ; Offset to start, 32-bit
```

```
                                                    ; (1000h:StartPM = StartPM + 10000h)
00000453 0800                       dw KCODE        ; This is the selector for CODE32_DESCRIPTOR,
                                                    ; assuming that StartPM resides in code32

                                [BITS 32]

                                StartPM:
                                    ; Kernel Base Address = 0 (Temporary) ; 30/12/2013
00000455 66B81000                   mov ax, KDATA       ; Save data segment identifier
00000459 8ED8                         mov ds, ax        ; Move a valid data segment into DS register
0000045B 8EC0                             mov es, ax        ; Move data segment into ES register
0000045D 8EE0                             mov fs, ax        ; Move data segment into FS register
0000045F 8EE8                             mov gs, ax        ; Move data segment into GS register
00000461 8ED0                         mov ss, ax        ; Move data segment into SS register
00000463 BC00000900                   mov esp, 90000h   ; Move the stack pointer to 090000h

                                memory_init:
                                    ; Initialize memory allocation table and page tables
                                    ; 16/11/2014
                                    ; 15/11/2014
                                    ; 07/11/2014
                                    ; 06/11/2014
                                    ; 05/11/2014
                                    ; 04/11/2014
                                    ; 31/10/2014 (Retro UNIX 386 v1 - Beginning)
                                    ;
00000468 31C0                       xor   eax, eax
0000046A 31C9                       xor   ecx, ecx
0000046C B108                       mov   cl, 8
0000046E BF00001000                 mov   edi, MEM_ALLOC_TBL
00000473 F3AB                       rep   stosd         ; clear Memory Allocation Table
                                                        ; for the first 1 MB memory
                                    ;
00000475 668B0D[1C490000]           mov   cx, [mem_1m_1k]       ; Number of contiguous KB between
                                                        ; 1 and 16 MB, max. 3C00h = 15 MB.
0000047C 66C1E902                   shr   cx, 2         ; convert 1 KB count to 4 KB count
00000480 890D[28490000]             mov   [free_pages], ecx
00000486 668B15[1E490000]           mov   dx, [mem_16m_64k]  ; Number of contiguous 64 KB blocks
                                                        ; between 16 MB and 4 GB.
0000048D 6609D2                     or    dx, dx
00000490 7413                       jz    short mi_0
                                    ;
00000492 6689D0                     mov   ax, dx
00000495 C1E004                     shl   eax, 4        ; 64 KB -> 4 KB (page count)
00000498 0105[28490000]             add   [free_pages], eax
0000049E 0500100000                 add   eax, 4096     ; 16 MB = 4096 pages
000004A3 EB07                       jmp   short mi_1
                                mi_0:
000004A5 6689C8                     mov   ax, cx
```

```
832 000004A8 66050001                   add    ax, 256               ; add 256 pages for the first 1 MB
833                                 mi_1:
834 000004AC A3[24490000]               mov    [memory_size], eax ; Total available memory in pages
835                                                                ; 1 alloc. tbl. bit = 1 memory page
836                                                                ; 32 allocation bits = 32 mem. pages
837                                     ;
838 000004B1 05FF7F0000                 add    eax, 32767      ; 32768 memory pages per 1 M.A.T. page
839 000004B6 C1E80F                     shr    eax, 15               ; ((32768 * x) + y) pages (y < 32768)
840                                                            ;  --> x + 1 M.A.T. pages, if y > 0
841                                                            ;  --> x M.A.T. pages, if y = 0
842 000004B9 66A3[38490000]             mov    [mat_size], ax        ; Memory Alloc. Table Size in pages
843 000004BF C1E00C                     shl    eax, 12               ; 1 M.A.T. page = 4096 bytes
844                                     ;                      ; Max. 32 M.A.T. pages (4 GB memory)
845 000004C2 89C3                       mov    ebx, eax        ; M.A.T. size in bytes
846                                     ; Set/Calculate Kernel's Page Directory Address
847 000004C4 81C300001000               add    ebx, MEM_ALLOC_TBL
848 000004CA 891D[20490000]             mov    [k_page_dir], ebx  ; Kernel's Page Directory address
849                                                                ; just after the last M.A.T. page
850                                     ;
851 000004D0 83E804                     sub    eax, 4          ; convert M.A.T. size to offset value
852 000004D3 A3[30490000]               mov    [last_page], eax   ; last page ofset in the M.A.T.
853                                     ;                      ; (allocation status search must be
854                                                            ; stopped after here)
855 000004D8 31C0                       xor    eax, eax
856 000004DA 48                         dec    eax             ; FFFFFFFFh (set all bits to 1)
857 000004DB 6651                       push   cx
858 000004DD C1E905                     shr    ecx, 5          ; convert 1 - 16 MB page count to
859                                                            ; count of 32 allocation bits
860 000004E0 F3AB                       rep    stosd
861 000004E2 6659                       pop    cx
862 000004E4 40                         inc    eax             ; 0
863 000004E5 80E11F                     and    cl, 31          ; remain bits
864 000004E8 7412                       jz     short mi_4
865 000004EA 8907                       mov    [edi], eax      ; reset
866                                 mi_2:
867 000004EC 0FAB07                     bts    [edi], eax      ; 06/11/2014
868 000004EF FEC9                       dec    cl
869 000004F1 7404                       jz     short mi_3
870 000004F3 FEC0                       inc    al
871 000004F5 EBF5                       jmp    short mi_2
872                                 mi_3:
873 000004F7 28C0                       sub    al, al          ; 0
874 000004F9 83C704                     add    edi, 4          ; 15/11/2014
875                                 mi_4:
876 000004FC 6609D2                     or     dx, dx         ; check 16M to 4G memory space
877 000004FF 7421                       jz     short mi_6     ; max. 16 MB memory, no more...
878                                     ;
879 00000501 B900021000                 mov    ecx, MEM_ALLOC_TBL + 512 ; End of first 16 MB memory
880                                     ;
```

```
881 00000506 29F9                       sub    ecx, edi       ; displacement (to end of 16 MB)
882 00000508 7406                       jz     short mi_5     ; jump if EDI points to
883                                                            ;         end of first 16 MB
884 0000050A D1E9                       shr    ecx, 1         ; convert to dword count
885 0000050C D1E9                       shr    ecx, 1         ; (shift 2 bits right)
886 0000050E F3AB                       rep    stosd          ; reset all bits for reserved pages
887                                                            ; (memory hole under 16 MB)
888                                 mi_5:
889 00000510 6689D1                     mov    cx, dx         ; count of 64 KB memory blocks
890 00000513 D1E9                       shr    ecx, 1         ; 1 alloc. dword per 128 KB memory
891 00000515 9C                         pushf                 ; 16/11/2014
892 00000516 48                         dec    eax            ; FFFFFFFFh (set all bits to 1)
893 00000517 F3AB                       rep    stosd
894 00000519 40                         inc    eax            ; 0
895 0000051A 9D                         popf                  ; 16/11/2014
896 0000051B 7305                       jnc    short mi_6
897 0000051D 6648                       dec    ax             ; eax = 0000FFFFh
898 0000051F AB                         stosd
899 00000520 6640                       inc    ax             ; 0
900                                 mi_6:
901 00000522 39DF                       cmp    edi, ebx       ; check if EDI points to
902 00000524 730A                       jnb    short mi_7     ; end of memory allocation table
903                                     ;                     ; (>= MEM_ALLOC_TBL + 4906)
904 00000526 89D9                       mov    ecx, ebx       ; end of memory allocation table
905 00000528 29F9                       sub    ecx, edi       ; convert displacement/offset
906 0000052A D1E9                       shr    ecx, 1         ; to dword count
907 0000052C D1E9                       shr    ecx, 1         ; (shift 2 bits right)
908 0000052E F3AB                       rep    stosd          ; reset all remain M.A.T. bits
909                                 mi_7:
910                                     ; Reset M.A.T. bits in M.A.T. (allocate M.A.T. pages)
911 00000530 BA00001000                 mov    edx, MEM_ALLOC_TBL
912                                     ;sub   ebx, edx       ; Mem. Alloc. Tbl. size in bytes
913                                     ;shr   ebx, 12              ; Mem. Alloc. Tbl. size in pages
914 00000535 668B0D[38490000]           mov    cx, [mat_size]       ; Mem. Alloc. Tbl. size in pages
915 0000053C 89D7                       mov    edi, edx
916 0000053E C1EF0F                     shr    edi, 15              ; convert M.A.T. address to
917                                                            ; byte offset in M.A.T.
918                                                            ; (1 M.A.T. byte points to
919                                                            ;         32768 bytes)
920                                                            ; Note: MEM_ALLOC_TBL address
921                                                            ; must be aligned on 128 KB
922                                                            ; boundary!
923 00000541 01D7                       add    edi, edx       ; points to M.A.T.'s itself
924                                     ; eax = 0
925 00000543 290D[28490000]             sub    [free_pages], ecx ; 07/11/2014
926                                 mi_8:
927 00000549 0FB307                     btr    [edi], eax     ; clear bit 0 to bit x (1 to 31)
928                                     ;dec   bl
929 0000054C FEC9                       dec    cl
```

```
930 0000054E 7404                       jz     short mi_9
931 00000550 FEC0                       inc    al
932 00000552 EBF5                       jmp    short mi_8
933                                 mi_9:
934                                     ;
935                                     ; Reset Kernel's Page Dir. and Page Table bits in M.A.T.
936                                     ;            (allocate pages for system page tables)
937
938                                     ; edx = MEM_ALLOC_TBL
939 00000554 8B0D[24490000]              mov    ecx, [memory_size] ; memory size in pages (PTEs)
940 0000055A 81C1FF030000              add    ecx, 1023     ; round up (1024 PTEs per table)
941 00000560 C1E90A                    shr    ecx, 10             ; convert memory page count to
942                                                          ; page table count (PDE count)
943                                     ;
944 00000563 51                        push   ecx           ; (**) PDE count (<= 1024)
945                                     ;
946 00000564 41                        inc    ecx           ; +1 for kernel page directory
947                                     ;
948 00000565 290D[28490000]            sub    [free_pages], ecx ; 07/11/2014
949                                     ;
950 0000056B 8B35[20490000]            mov    esi, [k_page_dir] ; Kernel's Page Directory address
951 00000571 C1EE0C                    shr    esi, 12             ; convert to page number
952                                 mi_10:
953 00000574 89F0                       mov    eax, esi      ; allocation bit offset
954 00000576 89C3                       mov    ebx, eax
955 00000578 C1EB03                    shr    ebx, 3        ; convert to alloc. byte offset
956 0000057B 80E3FC                    and    bl,  0FCh     ; clear bit 0 and bit 1
957                                                          ;   to align on dword boundary
958 0000057E 83E01F                    and    eax, 31             ; set allocation bit position
959                                                          ;  (bit 0 to bit 31)
960                                     ;
961 00000581 01D3                       add    ebx, edx      ; offset in M.A.T. + M.A.T. address
962                                     ;
963 00000583 0FB303                     btr    [ebx], eax    ; reset relevant bit (0 to 31)
964                                     ;
965 00000586 46                         inc    esi           ; next page table
966 00000587 E2EB                       loop   mi_10         ; allocate next kernel page table
967                                                          ; (ecx = page table count + 1)
968                                     ;
969 00000589 59                         pop    ecx           ; (**) PDE count (= pg. tbl. count)
970                                     ;
971                                     ; Initialize Kernel Page Directory and Kernel Page Tables
972                                     ;
973                                     ; Initialize Kernel's Page Directory
974 0000058A 8B3D[20490000]            mov    edi, [k_page_dir]
975 00000590 89F8                       mov    eax, edi
976 00000592 0C03                       or     al, PDE_A_PRESENT + PDE_A_WRITABLE
977                                                          ; supervisor + read&write + present
978 00000594 89CA                       mov    edx, ecx      ; (**) PDE count (= pg. tbl. count)
```

```
                                mi_11:
00000596 0500100000                   add   eax, 4096    ; Add page size (PGSZ)
                                                         ; EAX points to next page table
0000059B AB                           stosd
0000059C E2F8                         loop  mi_11
0000059E 29C0                         sub   eax, eax     ; Empty PDE
000005A0 66B90004                     mov   cx, 1024     ; Entry count (PGSZ/4)
000005A4 29D1                         sub   ecx, edx
000005A6 7402                         jz    short mi_12
000005A8 F3AB                         rep   stosd        ; clear remain (empty) PDEs
                                      ;
                                      ; Initialization of Kernel's Page Directory is OK, here.
                                mi_12:
                                      ; Initialize Kernel's Page Tables
                                      ;
                                      ; (EDI points to address of page table 0)
                                      ; eax = 0
000005AA 8B0D[24490000]               mov   ecx, [memory_size] ; memory size in pages
000005B0 89CA                         mov   edx, ecx     ; (***)
000005B2 B003                         mov   al, PTE_A_PRESENT + PTE_A_WRITABLE
                                                  ; supervisor + r214ead&write + present
                                mi_13:
000005B4 AB                           stosd
000005B5 0500100000                   add   eax, 4096
000005BA E2F8                         loop  mi_13
000005BC 6681E2FF03                   and   dx, 1023     ; (***)
000005C1 740B                         jz    short mi_14
000005C3 66B90004                     mov   cx, 1024
000005C7 6629D1                       sub   cx, dx       ; from dx (<= 1023) to 1024
000005CA 31C0                         xor   eax, eax
000005CC F3AB                         rep   stosd        ; clear remain (empty) PTEs
                                                         ; of the last page table
                                mi_14:
                                      ;  Initialization of Kernel's Page Tables is OK, here.
                                      ;
000005CE 89F8                         mov   eax, edi     ; end of the last page table page
                                                         ; (beginging of user space pages)
000005D0 C1E80F                       shr   eax, 15            ; convert to M.A.T. byte offset
000005D3 24FC                         and   al, 0FCh     ; clear bit 0 and bit 1 for
                                                         ; aligning on dword boundary
000005D5 A3[34490000]                 mov   [first_page], eax
000005DA A3[2C490000]                 mov   [next_page], eax ; The first free page pointer
                                                          ; for user programs
                                                          ; (Offset in Mem. Alloc. Tbl.)
                                      ;
                                      ; Linear/FLAT (1 to 1) memory paging for the kernel is OK, here.
                                      ;
```

```
1028                                    ; Enable paging
1029                                    ;
1030 000005DF A1[20490000]                mov     eax, [k_page_dir]
1031 000005E4 0F22D8                    mov    cr3, eax
1032 000005E7 0F20C0                    mov    eax, cr0
1033 000005EA 0D00000080                or     eax, 80000000h      ; set paging bit (bit 31)
1034 000005EF 0F22C0                    mov    cr0, eax
1035                                      ;jmp    KCODE:StartPMP
1036
1037 000005F2 EA                        db 0EAh             ; Opcode for far jump
1038 000005F3 [F9050000]                  dd StartPMP             ; 32 bit offset
1039 000005F7 0800                      dw KCODE            ; kernel code segment descriptor
1040
1041
1042                                  StartPMP:
1043                                    ; 06/11//2014
1044                                    ; Clear video page 0
1045                                    ;
1046                                    ; Temporary Code
1047                                    ;
1048 000005F9 B9E8030000                mov    ecx, 80*25/2
1049 000005FE BF00800B00                mov    edi, 0B8000h
1050 00000603 31C0                      xor    eax, eax     ; black background, black fore color
1051 00000605 F3AB                      rep    stosd
1052
1053                                    ; 19/08/2014
1054                                    ; Kernel Base Address = 0
1055                                    ; It is mapped to (physically) 0 in the page table.
1056                                    ; So, here is exactly 'StartPMP' address.
1057                                    ;
1058 00000607 B44E                      mov    ah, 4Eh      ; Red background, yellow forecolor
1059 00000609 BE[3E0E0000]              mov    esi, msgPM
1060 0000060E BF00800B00                mov    edi, 0B8000h ; 27/08/2014
1061                                    ; 20/08/2014
1062 00000613 E8F9000000                call   printk
1063
1064                                    ; 'UNIX v7/x86' source code by Robert Nordier (1999)
1065                                    ; // Set IRQ offsets
1066                                    ;
1067                                    ;  Linux (v0.12) source code by Linus Torvalds (1991)
1068                                    ;
1069                                                                ;; ICW1
1070 00000618 B011                      mov    al, 11h                  ; Initialization sequence
1071 0000061A E620                      out    20h, al                  ;      8259A-1
1072                                    ; jmp  $+2
1073 0000061C E6A0                      out    0A0h, al             ;      8259A-2
1074                                                                ;; ICW2
1075 0000061E B020                      mov    al, 20h                  ; Start of hardware ints (20h)
1076 00000620 E621                      out    21h, al                  ;      for 8259A-1
```

```
1077                                    ; jmp  $+2
1078 00000622 B028                      mov    al, 28h                  ; Start of hardware ints (28h)
1079 00000624 E6A1                      out    0A1h, al           ;     for 8259A-2
1080                                                               ;
1081 00000626 B004                      mov    al, 04h                  ;; ICW3
1082 00000628 E621                      out    21h, al                  ;     IRQ2 of 8259A-1 (master)
1083                                    ; jmp  $+2
1084 0000062A B002                      mov    al, 02h                  ;     is 8259A-2 (slave)
1085 0000062C E6A1                      out    0A1h, al           ;
1086                                                               ;; ICW4
1087 0000062E B001                      mov    al, 01h                  ;
1088 00000630 E621                      out    21h, al                  ;     8086 mode, normal EOI
1089                                    ; jmp  $+2
1090 00000632 E6A1                      out    0A1h, al           ;     for both chips.
1091
1092                                    ;mov   al, 0FFh     ; mask off all interrupts for now
1093                                    ;out   21h, al
1094                                    ;; jmp       $+2
1095                                    ;out   0A1h, al
1096
1097                                    ;
1098                                    ;; Linux (v0.12) source code by Linus Torvalds (1991)
1099                                    ;  setup_idt:
1100                                    ;
1101                                      ;; 16/02/2015
1102                                    ;;mov     dword [DISKETTE_INT], fdc_int ; IRQ 6 handler
1103                                    ; 21/08/2014 (timer_int)
1104 00000634 BE[1B070000]              mov    esi, ilist
1105 00000639 8D3D[300C0000]            lea    edi, [idt]
1106 0000063F B940000000                mov    ecx, 64             ; 64 interrupts (INT 0 to INT 3Fh)
1107                                  rp_sidt1:
1108 00000644 AD                        lodsd
1109 00000645 09C0                      or     eax, eax
1110 00000647 740E                      jz     short rp_sidt2
1111 00000649 89C2                      mov    edx, eax
1112 0000064B BB00000800                mov    ebx, 80000h
1113 00000650 6689D3                      mov     bx, dx          ; /* selector = 0x0008 = cs */
1114 00000653 66BA008E                    mov     dx, 8E00h       ; /* interrupt gate - dpl=0, present */
1115                                  rp_sidt2:
1116 00000657 891F                      mov    [edi], ebx
1117 00000659 895704                    mov    [edi+4], edx
1118 0000065C 83C708                    add    edi, 8
1119 0000065F 49                        dec    ecx
1120 00000660 7406                      jz     short sidt_OK
1121 00000662 21C0                      and    eax, eax
1122 00000664 74F1                      jz     short rp_sidt2
1123 00000666 EBDC                      jmp    short rp_sidt1
1124                                  sidt_OK:
1125 00000668 0F011D[360E0000]          lidt   [idtd]
```

```
1126                                    ;
1127                                    ; 10/11/2014 (Retro UNIX 386 v1 - Erdogan Tan)
1128                                    ;
1129                                    ;cli   ; Disable interrupts
1130                                    ;    (CPU will not handle hardware interrupts, except NMI!)
1131                                    ;
1132 0000066F 30C0                       xor    al, al       ; Enable all hardware interrupts!
1133 00000671 E621                     out    21h, al             ; (IBM PC-AT compatibility)
1134 00000673 EB00                     jmp    $+2          ; (All conventional PC-AT hardware
1135 00000675 E6A1                     out    0A1h, al     ;  interrupts will be in use.)
1136                                                        ; (Even if related hardware component
1137                                                        ;  does not exist!)
1138                                    ; Enable NMI
1139 00000677 B07F                     mov    al, 7Fh             ; Clear bit 7 to enable NMI (again)
1140 00000679 E670                     out    70h, al
1141                                    ; 23/02/2015
1142 0000067B 90                       nop
1143 0000067C E471                     in     al, 71h             ; read in 71h just after writing out to 70h
1144                                                        ; for preventing unknown state (!?)
1145                                    ;
1146                                    ; Only a NMI can occur here... (Before a 'STI' instruction)
1147                                    ;
1148                                    ; 02/09/2014
1149 0000067E 6631DB                   xor    bx, bx
1150 00000681 66BA0003                 mov    dx, 0300h    ; Row 3, column 0
1151 00000685 E8E0130000               call   set_cpos
1152                                    ;
1153                                    ; 06/11/2014
1154                                    ; Temporary code
1155                                    ;
1156 0000068A E8F8150000               call   memory_info
1157                                    ;
1158 0000068F E841320000               call   getch  ; 28/02/2015
1159                                drv_init:
1160 00000694 FB                       sti    ; Enable Interrupts
1161                                    ; 28/02/2015
1162 00000695 66A1[3C490000]           mov    ax, [cursor_posn] ; cursor pos. for video page 0
1163 0000069B 66A3[20480000]           mov    [cursor_posb], ax ; cursor position backup
1164                                    ;
1165                                    ; 06/02/2015
1166 000006A1 8B15[18490000]           mov    edx, [hd0_type] ; hd0, hd1, hd2, hd3
1167 000006A7 668B1D[16490000]         mov    bx, [fd0_type] ; fd0, fd1
1168                                    ; 22/02/2015
1169 000006AE 6621DB                   and    bx, bx
1170 000006B1 7531                     jnz    short di1
1171                                    ;
1172 000006B3 09D2                     or     edx, edx
1173 000006B5 753F                     jnz    short di2
1174                                    ;
```

```
1175                                    setup_error:
1176 000006B7 BE[CE060000]                     mov   esi, setup_error_msg
1177                                    psem:
1178 000006BC AC                               lodsb
1179 000006BD 08C0                             or    al, al
1180                                           ;jz   short hang ; 22/02/2015
1181 000006BF 743C                             jz    short di3
1182 000006C1 56                               push  esi
1183 000006C2 31DB                             xor   ebx, ebx ; 0
1184                                                       ; Video page 0 (bl=0)
1185 000006C4 B407                             mov   ah, 07h ; Black background,
1186                                                       ; light gray forecolor
1187 000006C6 E871120000                       call  write_tty
1188 000006CB 5E                               pop   esi
1189 000006CC EBEE                             jmp   short psem
1190
1191                                    setup_error_msg:
1192 000006CE 0D0A                             db 0Dh, 0Ah
1193 000006D0 4469736B2053657475-              db 'Disk Setup Error!'
1194 000006D9 70204572726F7221
1195 000006E1 0D0A00                           db 0Dh, 0Ah,0
1196                                    di1:
1197                                           ; supress 'jmp short T6'
1198                                           ;  (activate fdc motor control code)
1199 000006E4 66C705[26080000]90-              mov   word [T5], 9090h ; nop
1200 000006EC 90
1201                                           ;
1202                                           ;mov   ax, int_0Eh  ; IRQ 6 handler
1203                                           ;mov   di, 0Eh*4    ; IRQ 6 vector
1204                                           ;stosw
1205                                           ;mov   ax, cs
1206                                           ;stosw
1207                                           ;; 16/02/2015
1208                                             ;;mov     dword [DISKETTE_INT], fdc_int ; IRQ 6 handler
1209                                           ;
1210 000006ED E878250000                       CALL  DSKETTE_SETUP; Initialize Floppy Disks
1211                                           ;
1212 000006F2 09D2                             or    edx, edx
1213 000006F4 7407                               jz      short di3
1214                                    di2:
1215 000006F6 E8D1250000                       call        DISK_SETUP   ; Initialize Fixed Disks
1216 000006FB 72BA                               jc      short setup_error
1217                                    di3:
1218 000006FD E8FE2D0000                       call  setup_rtc_int; 22/05/2015 (dsectrpm.s)
1219                                           ;
1220                                           ;call  dsectpm  ; 06/02/2015 - 21/02/2015
1221 00000702 E8252E0000                       call  display_sectors ; 22/02/2015
1222                                    hang:
1223 00000707 E8C9310000                       call  getch ; 22/02/2015
```

```
                                        ;
0000070C E9360D0000                     jmp   cpu_reset ; 22/02/2015

                                        ; 27/08/2014
                                        ; 20/08/2014
                                printk:
                                          ;mov    edi, dword [scr_row]
                                pkl:
00000711 AC                             lodsb
00000712 08C0                           or    al, al
00000714 7404                           jz    short pkr
00000716 66AB                           stosw
00000718 EBF7                           jmp   short pkl
                                pkr:
0000071A C3                             retn

                                        ; 21/08/2014
                                ilist:
                                        ;times       32 dd cpu_except ; INT 0 to INT 1Fh
                                        ;
                                        ; Exception list
                                        ; 25/08/2014
0000071B [FB080000]                     dd    exc0   ; 0h,  Divide-by-zero Error
0000071F [02090000]                     dd    exc1
00000723 [09090000]                     dd    exc2
00000727 [10090000]                     dd    exc3
0000072B [17090000]                     dd    exc4
0000072F [1E090000]                     dd    exc5
00000733 [25090000]                     dd    exc6   ; 06h,  Invalid Opcode
00000737 [2C090000]                     dd    exc7
0000073B [30090000]                     dd    exc8
0000073F [34090000]                     dd    exc9
00000743 [38090000]                     dd    exc10
00000747 [3C090000]                     dd    exc11
0000074B [40090000]                     dd    exc12
0000074F [44090000]                     dd    exc13  ; 0Dh, General Protection Fault
00000753 [48090000]                     dd    exc14  ; 0Eh, Page Fault
00000757 [4C090000]                     dd    exc15
0000075B [50090000]                     dd    exc16
0000075F [54090000]                     dd    exc17
00000763 [58090000]                     dd    exc18
00000767 [5C090000]                     dd    exc19
0000076B [60090000]                     dd    exc20
0000076F [64090000]                     dd    exc21
00000773 [68090000]                     dd    exc22
00000777 [6C090000]                     dd    exc23
0000077B [70090000]                     dd    exc24
0000077F [74090000]                     dd    exc25
00000783 [78090000]                     dd    exc26
```

```
1273 00000787 [7C090000]                dd     exc27
1274 0000078B [80090000]                dd     exc28
1275 0000078F [84090000]                dd     exc29
1276 00000793 [88090000]                dd     exc30
1277 00000797 [8C090000]                dd     exc31
1278                                    ; Interrupt list
1279 0000079B [E3070000]                dd     timer_int    ; INT 20h
1280                                           ;dd    irq0
1281                                    ;dd    keyb_int     ; 27/08/2014
1282 0000079F [54110000]                dd     kb_int       ; 22/02/2015 - dsectpm.s
1283                                           ;dd    irq1
1284 000007A3 [60080000]                dd     irq2
1285 000007A7 [64080000]                dd     irq3
1286 000007AB [68080000]                dd     irq4
1287 000007AF [6C080000]                dd     irq5
1288                                  ;DISKETTE_INT: ;06/02/2015
1289 000007B3 [5A2C0000]                dd     fdc_int            ; 16/02/2015, IRQ 6 handler
1290                                           ;dd    irq6
1291                                  ; Default IRQ 7 handler against spurious IRQs (from master PIC)
1292                                  ; 25/02/2015 (source: http://wiki.osdev.org/8259_PIC)
1293 000007B7 [B70B0000]                dd     default_irq7 ; 25/02/2015
1294                                           ;dd    irq7
1295                                  ; Real Time Clock Interrupt
1296 000007BB [690A0000]                dd     rtc_int            ; 22/02/2015 - IRQ 8 handler - dsectpm.s
1297                                           ;dd    irq8   ; INT 28h
1298 000007BF [7C080000]                dd     irq9
1299 000007C3 [80080000]                dd     irq10
1300 000007C7 [84080000]                dd     irq11
1301 000007CB [88080000]                dd     irq12
1302 000007CF [8C080000]                dd     irq13
1303                                  ;HDISK_INT1:  ;06/02/2015
1304 000007D3 [B3340000]                dd     hdc1_int     ; 21/02/2015, IRQ 14 handler
1305                                           ;dd    irq14
1306                                  ;HDISK_INT2:  ;06/02/2015
1307 000007D7 [DA340000]                dd     hdc2_int     ; 21/02/2015, IRQ 15 handler
1308                                           ;dd    irq15  ; INT 2Fh
1309 000007DB [2F0A0000]                dd     ignore_int
1310 000007DF 00000000                  dd     0
1311
1312                                  ; 17/02/2015
1313                                  ; 06/02/2015 (unix386.s)
1314                                  ; 11/12/2014 - 22/12/2014 (dsectrm2.s)
1315                                  ;
1316                                  ; IBM PC-XT Model 286 Source Code - BIOS2.ASM (06/10/85)
1317                                  ;
1318                                  ;-- HARDWARE INT  08 H - ( IRQ LEVEL 0 ) ----------------------------------------
1319                                  ;     THIS ROUTINE HANDLES THE TIMER INTERRUPT FROM FROM CHANNEL 0 OF        :
1320                                  ;     THE 8254 TIMER.  INPUT FREQUENCY IS 1.19318 MHZ AND THE DIVISOR        :
1321                                  ;     IS 65536, RESULTING IN APPROXIMATELY 18.2 INTERRUPTS EVERY SECOND.     :
```

```
                                ;                                                                      :
                                ;     THE INTERRUPT HANDLER MAINTAINS A COUNT (40:6C) OF INTERRUPTS SINCE    :
                                ;     POWER ON TIME, WHICH MAY BE USED TO ESTABLISH TIME OF DAY.        :
                                ;     THE INTERRUPT HANDLER ALSO DECREMENTS THE MOTOR CONTROL COUNT (40:40)  :
                                ;     OF THE DISKETTE, AND WHEN IT EXPIRES, WILL TURN OFF THE           :
                                ;     DISKETTE MOTOR(s), AND RESET THE MOTOR RUNNING FLAGS.                   :
                                ;     THE INTERRUPT HANDLER WILL ALSO INVOKE A USER ROUTINE THROUGH           :
                                ;     INTERRUPT 1CH AT EVERY TIME TICK.  THE USER MUST CODE A           :
                                ;     ROUTINE AND PLACE THE CORRECT ADDRESS IN THE VECTOR TABLE.        :
                                ;--------------------------------------------------------------------------
                                ;

                                timer_int:  ; IRQ 0
                                ;int_08h:   ; Timer
000007E3 FB                           STI                       ; INTERRUPTS BACK ON
                                      ;PUSH  DS
                                      ;PUSH  AX
                                      ;PUSH  DX                 ; SAVE MACHINE STATE
                                      ;xor   ax,ax ; real mode (dsectrm2.s)
                                      ;mov   ds,ax ; real mode (dsectrm2.s)
                                      ;
000007E4 50                           push   eax
000007E5 52                           push   edx
000007E6 51                           push   ecx
000007E7 53                           push   ebx
000007E8 1E                           push   ds
000007E9 06                           push   es
000007EA B810000000                   mov    eax, KDATA
000007EF 8ED8                         mov    ds, ax
000007F1 8EC0                         mov    es, ax
                                      ;
000007F3 66FF05[CD1D0000]             INC    word [TIMER_LOW]   ; INCREMENT TIME
000007FA 7507                         JNZ    short T4           ; GO TO TEST_DAY
000007FC 66FF05[CF1D0000]             INC    word [TIMER_HIGH]  ; INCREMENT HIGH WORD OF TIME
                                T4:                             ; TEST_DAY
00000803 66833D[CF1D0000]18           CMP    word [TIMER_HIGH],018H    ; TEST FOR COUNT EQUALING 24 HOURS
0000080B 7519                         JNZ    short T5           ; GO TO DISKETTE_CTL
0000080D 66813D[CD1D0000]B0-          CMP    word [TIMER_LOW],0B0H
00000815 00
00000816 750E                         JNZ    short T5           ; GO TO DISKETTE_CTL

                                ;-----      TIMER HAS GONE 24 HOURS
                                      ;;SUB  AX,AX
                                      ;MOV   word [TIMER_HIGH],AX
                                      ;MOV   word [TIMER_LOW],AX
00000818 29C0                         sub    eax, eax
0000081A A3[CD1D0000]                 mov    dword [TIMER_LH], eax
                                      ;
0000081F C605[D11D0000]01             MOV    byte [TIMER_OFL],1
```

```
1371
1372                                    ;-----      TEST FOR DISKETTE TIME OUT
1373
1374                                    T5:
1375                                          ; 23/12/2014
1376 00000826 EB1D                            jmp   short T6            ; will be replaced with nop, nop
1377                                                                    ; (9090h) if a floppy disk
1378                                                                    ; is detected.
1379                                          ;mov  al,[CS:MOTOR_COUNT]
1380 00000828 A0[C41D0000]                    mov   al, [MOTOR_COUNT]
1381 0000082D FEC8                            dec   al
1382                                          ;mov  [CS:MOTOR_COUNT], al      ; DECREMENT DISKETTE MOTOR CONTROL
1383 0000082F A2[C41D0000]                    mov   [MOTOR_COUNT], al
1384                                          ;mov  [ORG_MOTOR_COUNT], al
1385 00000834 750F                            JNZ   short T6            ; RETURN IF COUNT NOT OUT
1386 00000836 B0F0                            mov   al,0F0h
1387                                          ;AND  [CS:MOTOR_STATUS],al      ; TURN OFF MOTOR RUNNING BITS
1388 00000838 2005[C31D0000]                  and   [MOTOR_STATUS], al
1389                                          ;and  [ORG_MOTOR_STATUS], al
1390 0000083E B00C                            MOV   AL,0CH              ; bit 3 = enable IRQ & DMA,
1391                                                                    ; bit 2 = enable controller
1392                                                                    ;      1 = normal operation
1393                                                                    ;      0 = reset
1394                                                                    ; bit 0, 1 = drive select
1395                                                                    ; bit 4-7 = motor running bits
1396 00000840 66BAF203                        MOV   DX,03F2H            ; FDC CTL PORT
1397 00000844 EE                              OUT   DX,AL               ; TURN OFF THE MOTOR
1398                                    T6:
1399                                          ;inc  word [CS:wait_count]      ; 22/12/2014 (byte -> word)
1400                                                                    ; TIMER TICK INTERRUPT
1401 00000845 66FF05[2A480000]                inc   word [wait_count]
1402                                          ;
1403                                          ;; DS = 0 (DS <> CS) !
1404                                          ;INT  1CH                 ; TRANSFER CONTROL TO A USER ROUTINE
1405                                          ;;;;cli      ; 17/02/2015
1406                                          ;; 22/02/2015 (dsectpm.s --> cancel 'u_timer' call)
1407                                          ;;call       u_timer                    ; TRANSFER CONTROL TO A USER ROUTINE
1408                                          ;
1409                                          ;POP  DX                  ; RESTORE (DX)
1410 0000084C B020                            MOV   AL,EOI              ; GET END OF INTERRUPT MASK
1411 0000084E FA                              CLI                       ; DISABLE INTERRUPTS TILL STACK CLEARED
1412 0000084F E620                            OUT   INTA00,AL           ; END OF INTERRUPT TO 8259 - 1
1413                                          ;POP  AX
1414                                          ;POP  DS                  ; RESET MACHINE STATE
1415                                          ;
1416 00000851 07                              pop   es
1417 00000852 1F                              pop   ds
1418 00000853 5B                              pop   ebx
1419 00000854 59                              pop   ecx
```

```
1420 00000855 5A                        pop    edx
1421 00000856 58                        pop    eax
1422                                    ;
1423 00000857 CF                        IRETd                   ; RETURN FROM INTERRUPT
1424
1425                                ; ///////////////
1426
1427                                    ; 23/02/2015
1428                                    ; 06/02/2015
1429                                    ; 07/09/2014
1430                                    ; 21/08/2014
1431                                ;u_timer:
1432                                ;timer_int:  ; IRQ 0
1433                                    ; 06/02/2015
1434                                    ;push  eax
1435                                    ;push  edx
1436                                    ;push  ecx
1437                                    ;push  ebx
1438                                    ;push  ds
1439                                    ;push  es
1440                                    ;mov   eax, KDATA
1441                                    ;mov   ds, ax
1442                                    ;mov   es, ax
1443                                ;   inc    dword [tcount]
1444                                ;   mov    ebx, tcountstr + 4
1445                                ;   mov    ax, word [tcount]
1446                                ;   mov    ecx, 10
1447                                ;rp_divtcnt:
1448                                ;   xor    edx, edx
1449                                ;   div    ecx
1450                                ;   add    dl, 30h
1451                                ;   mov    byte [ebx], dl
1452                                ;   or     ax, ax
1453                                ;   jz     short print_lzero
1454                                ;   dec    ebx
1455                                ;   jmp    short rp_divtcnt
1456                                ;print_lzero:
1457                                ;   cmp    ebx, tcountstr
1458                                ;   jna    short print_tcount
1459                                ;   dec    ebx
1460                                 ;   mov    byte [ebx], 30h
1461                                ;   jmp    short print_lzero
1462                                ;print_tcount:
1463                                ;   push   esi
1464                                ;   push   edi
1465                                ;   mov    esi, timer_msg ; Timer interrupt message
1466                                    ; 07/09/2014
1467                                ;   mov    bx, 1        ; Video page 1
1468                                ;ptmsg:
```

```
1469                                    ;       lodsb
1470                                    ;       or      al, al
1471                                    ;       jz      short ptmsg_ok
1472                                    ;       push    esi
1473                                    ;       push    bx
1474                                    ;         mov     ah,  2Fh ; Green background, white forecolor
1475                                    ;       call    write_tty
1476                                    ;       pop     bx
1477                                    ;       pop     esi
1478                                    ;       jmp     short ptmsg
1479                                            ;; 27/08/2014
1480                                            ;mov     edi, 0B8000h + 0A0h ; Row 1
1481                                            ;call  printk
1482                                            ;
1483                                    ;ptmsg_ok:
1484                                            ; 07/09/2014
1485                                    ;       xor     dx, dx          ; column 0, row 0
1486                                    ;       call    set_cpos        ; set cursor position to 0,0
1487                                            ;; 23/02/2015
1488                                            ; 25/08/2014
1489                                    ;;      mov     ebx, scounter           ; (seconds counter)
1490                                    ;;      dec     byte [ebx+1]            ; (for reading real time clock)
1491                                    ;;;     jns     short timer_eoi                 ; 0 -> 0FFh ?
1492                                    ;;      jns     short u_timer_retn
1493                                    ;;      mov     byte [ebx+1], 18        ; (18.2 timer ticks per second)
1494                                    ;;      dec     byte [ebx]              ; 19+18+18+18+18 (5)
1495                                    ;;;     jnz     short timer_eoi                 ; (109 timer ticks in 5 seconds)
1496                                    ;;      jnz     short u_timer_retn ; 06/02/2015
1497                                    ;;      mov     byte [ebx], 5
1498                                    ;;      inc     byte [ebx+1] ; 19
1499                                    ;;;timer_eoi:
1500                                    ;;;     mov     al, 20h ; END OF INTERRUPT COMMAND TO 8259
1501                                    ;;;     out     20h, al         ; 8259 PORT
1502                                    ;;      ;
1503                                    ;;u_timer_retn:  ; 06/02/2015
1504                                    ;       pop     edi
1505                                    ;       pop     esi
1506                                    ;;      ;pop    es
1507                                    ;;      ;pop    ds
1508                                    ;;      ;pop    ebx
1509                                    ;;      ;pop    ecx
1510                                    ;;      ;pop    edx
1511                                    ;;      ;pop    eax
1512                                    ;;      ;iret
1513                                    ;       retn    ; 06/02/2015
1514
1515                                            ; 28/08/2014
1516                                    irq0:
1517 00000858 6A00                              push       dword 0
```

```
1518 0000085A EB3A                      jmp    short which_irq
1519                                irq1:
1520 0000085C 6A01                        push       dword 1
1521 0000085E EB36                      jmp    short which_irq
1522                                irq2:
1523 00000860 6A02                        push       dword 2
1524 00000862 EB32                      jmp    short which_irq
1525                                irq3:
1526 00000864 6A03                        push       dword 3
1527 00000866 EB2E                      jmp    short which_irq
1528                                irq4:
1529 00000868 6A04                        push       dword 4
1530 0000086A EB2A                      jmp    short which_irq
1531                                irq5:
1532 0000086C 6A05                        push       dword 5
1533 0000086E EB26                      jmp    short which_irq
1534                                irq6:
1535 00000870 6A06                        push       dword 6
1536 00000872 EB22                      jmp    short which_irq
1537                                irq7:
1538 00000874 6A07                        push       dword 7
1539 00000876 EB1E                      jmp    short which_irq
1540                                irq8:
1541 00000878 6A08                        push       dword 8
1542 0000087A EB1A                      jmp    short which_irq
1543                                irq9:
1544 0000087C 6A09                        push       dword 9
1545 0000087E EB16                      jmp    short which_irq
1546                                irq10:
1547 00000880 6A0A                        push       dword 10
1548 00000882 EB12                      jmp    short which_irq
1549                                irq11:
1550 00000884 6A0B                        push       dword 11
1551 00000886 EB0E                      jmp    short which_irq
1552                                irq12:
1553 00000888 6A0C                        push       dword 12
1554 0000088A EB0A                      jmp    short which_irq
1555                                irq13:
1556 0000088C 6A0D                        push       dword 13
1557 0000088E EB06                      jmp    short which_irq
1558                                irq14:
1559 00000890 6A0E                        push       dword 14
1560 00000892 EB02                      jmp    short which_irq
1561                                irq15:
1562 00000894 6A0F                        push       dword 15
1563                                    ;jmp   short which_irq
1564
1565
1566                                    ; 29/08/2014
```

```
1567                                    ; 21/08/2014
1568                                which_irq:
1569 00000896 870424                    xchg   eax, [esp]  ; 28/08/2014
1570 00000899 53                        push   ebx
1571 0000089A 56                        push   esi
1572 0000089B 57                        push   edi
1573 0000089C 1E                        push   ds
1574 0000089D 06                        push   es
1575                                    ;
1576 0000089E 88C3                      mov    bl, al
1577                                    ;
1578 000008A0 B810000000                mov    eax, KDATA
1579 000008A5 8ED8                      mov    ds, ax
1580 000008A7 8EC0                      mov    es, ax
1581
1582                                      ; 27/08/2014
1583 000008A9 8105[0A0C0000]A000-         add     dword [scr_row], 0A0h
1584 000008B1 0000
1585                                    ;
1586 000008B3 B417                      mov    ah, 17h      ; blue (1) background,
1587                                                  ; light gray (7) forecolor
1588 000008B5 8B3D[0A0C0000]              mov     edi, [scr_row]
1589 000008BB B049                      mov    al, 'I'
1590 000008BD 66AB                      stosw
1591 000008BF B052                      mov    al, 'R'
1592 000008C1 66AB                      stosw
1593 000008C3 B051                      mov    al, 'Q'
1594 000008C5 66AB                      stosw
1595 000008C7 B020                      mov    al, ' '
1596 000008C9 66AB                      stosw
1597 000008CB 88D8                      mov    al, bl
1598 000008CD 3C0A                      cmp    al, 10
1599 000008CF 7208                      jb     short iix
1600 000008D1 B031                      mov    al, '1'
1601 000008D3 66AB                      stosw
1602 000008D5 88D8                      mov    al, bl
1603 000008D7 2C0A                      sub    al, 10
1604                                iix:
1605 000008D9 0430                      add    al, '0'
1606 000008DB 66AB                      stosw
1607 000008DD B020                      mov    al, ' '
1608 000008DF 66AB                      stosw
1609 000008E1 B021                      mov    al, '!'
1610 000008E3 66AB                      stosw
1611 000008E5 B020                      mov    al, ' '
1612 000008E7 66AB                      stosw
1613                                    ; 23/02/2015
1614 000008E9 80FB07                    cmp    bl, 7 ; check for IRQ 8 to IRQ 15
1615 000008EC 0F866C010000              jna    iiret
```

```
1616 000008F2 B020                      mov   al, 20h  ; END OF INTERRUPT COMMAND TO
1617 000008F4 E6A0                      out   0A0h, al ; the 2nd 8259
1618 000008F6 E963010000                jmp    iiret
1619                                    ;
1620                                    ; 22/08/2014
1621                                    ;mov  al, 20h ; END OF INTERRUPT COMMAND TO 8259
1622                                    ;out  20h, al      ; 8259 PORT
1623                                    ;
1624                                    ;pop  es
1625                                    ;pop  ds
1626                                    ;pop  edi
1627                                    ;pop  esi
1628                                    ;pop  ebx
1629                                    ;pop  eax
1630                                    ;iret
1631
1632                                    ; 25/08/2014
1633                                exc0:
1634 000008FB 6A00                        push      dword 0
1635 000008FD E9AA000000                  jmp    cpu_except
1636                                exc1:
1637 00000902 6A01                        push      dword 1
1638 00000904 E9A3000000                  jmp    cpu_except
1639                                exc2:
1640 00000909 6A02                        push      dword 2
1641 0000090B E99C000000                  jmp    cpu_except
1642                                exc3:
1643 00000910 6A03                        push      dword 3
1644 00000912 E995000000                  jmp    cpu_except
1645                                exc4:
1646 00000917 6A04                        push      dword 4
1647 00000919 E98E000000                  jmp    cpu_except
1648                                exc5:
1649 0000091E 6A05                        push      dword 5
1650 00000920 E987000000                  jmp    cpu_except
1651                                exc6:
1652 00000925 6A06                        push      dword 6
1653 00000927 E980000000                  jmp    cpu_except
1654                                exc7:
1655 0000092C 6A07                        push      dword 7
1656 0000092E EB7C                        jmp    cpu_except
1657                                exc8:
1658                                    ; [esp] = Error code
1659 00000930 6A08                        push      dword 8
1660 00000932 EB5A                        jmp    cpu_except_en
1661                                exc9:
1662 00000934 6A09                        push      dword 9
1663 00000936 EB74                        jmp    cpu_except
1664                                exc10:
```

```
1665                                    ; [esp] = Error code
1666 00000938 6A0A                       push       dword 10
1667 0000093A EB52                       jmp     cpu_except_en
1668                                exc11:
1669                                    ; [esp] = Error code
1670 0000093C 6A0B                       push       dword 11
1671 0000093E EB4E                       jmp     cpu_except_en
1672                                exc12:
1673                                    ; [esp] = Error code
1674 00000940 6A0C                       push       dword 12
1675 00000942 EB4A                       jmp     cpu_except_en
1676                                exc13:
1677                                    ; [esp] = Error code
1678 00000944 6A0D                       push       dword 13
1679 00000946 EB46                       jmp     cpu_except_en
1680                                exc14:
1681                                    ; [esp] = Error code
1682 00000948 6A0E                       push       dword 14
1683 0000094A EB42                     jmp    short cpu_except_en
1684                                exc15:
1685 0000094C 6A0F                       push       dword 15
1686 0000094E EB5C                       jmp     cpu_except
1687                                exc16:
1688 00000950 6A10                       push       dword 16
1689 00000952 EB58                       jmp     cpu_except
1690                                exc17:
1691                                    ; [esp] = Error code
1692 00000954 6A11                       push       dword 17
1693 00000956 EB36                     jmp    short cpu_except_en
1694                                exc18:
1695 00000958 6A12                       push       dword 18
1696 0000095A EB50                     jmp    short cpu_except
1697                                exc19:
1698 0000095C 6A13                       push       dword 19
1699 0000095E EB4C                     jmp    short cpu_except
1700                                exc20:
1701 00000960 6A14                       push       dword 20
1702 00000962 EB48                     jmp    short cpu_except
1703                                exc21:
1704 00000964 6A15                       push       dword 21
1705 00000966 EB44                     jmp    short cpu_except
1706                                exc22:
1707 00000968 6A16                       push       dword 22
1708 0000096A EB40                     jmp    short cpu_except
1709                                exc23:
1710 0000096C 6A17                       push       dword 23
1711 0000096E EB3C                     jmp    short cpu_except
1712                                exc24:
1713 00000970 6A18                       push       dword 24
```

```
1714 00000972 EB38                      jmp    short cpu_except
1715                                    exc25:
1716 00000974 6A19                        push       dword 25
1717 00000976 EB34                      jmp    short cpu_except
1718                                    exc26:
1719 00000978 6A1A                        push       dword 26
1720 0000097A EB30                      jmp    short cpu_except
1721                                    exc27:
1722 0000097C 6A1B                        push       dword 27
1723 0000097E EB2C                      jmp    short cpu_except
1724                                    exc28:
1725 00000980 6A1C                        push       dword 28
1726 00000982 EB28                      jmp    short cpu_except
1727                                    exc29:
1728 00000984 6A1D                        push       dword 29
1729 00000986 EB24                      jmp    short cpu_except
1730                                    exc30:
1731 00000988 6A1E                        push       dword 30
1732 0000098A EB20                      jmp    short cpu_except
1733                                    exc31:
1734 0000098C 6A1F                        push       dword 31
1735                                      ;jmp     short cpu_except
1736
1737                                    ; 28/08/2014
1738                                    cpu_except_en:
1739 0000098E 87442404                  xchg   eax, [esp+4] ; Error code
1740 00000992 1E                        push   ds
1741 00000993 6650                      push   ax
1742 00000995 66B81000                  mov    ax, KDATA
1743 00000999 8ED8                      mov    ds, ax
1744 0000099B 6658                      pop    ax
1745 0000099D A3[34190000]              mov    dword [error_code], eax
1746 000009A2 1F                        pop    ds
1747 000009A3 8B0424                    mov    eax, [esp] ; Exception number
1748 000009A6 83C404                    add    esp, 4
1749 000009A9 870424                    xchg   eax, [esp]
1750                                           ; eax = eax before exception
1751                                           ; [esp] -> exception number
1752                                           ; [esp+4] -> EIP to return
1753                                    ; 29/08/2014
1754                                    ; 28/08/2014
1755                                    ; 25/08/2014
1756                                    ; 21/08/2014
1757                                    cpu_except:        ; CPU Exceptions
1758 000009AC FC                        cld
1759 000009AD 870424                    xchg   eax, [esp]
1760                                                ; eax = Exception number
1761                                                ; [esp] = eax (before exception)
1762 000009B0 53                        push   ebx
```

```
1763 000009B1 BB[07070000]              mov   ebx, hang
1764 000009B6 875C2408                  xchg  ebx, [esp+8]
1765                                                ; EIP (points to instruction which faults)
1766                                                ; New EIP (hang)
1767 000009BA 56                        push  esi
1768 000009BB 57                        push  edi
1769 000009BC 1E                        push  ds
1770 000009BD 06                        push  es
1771                                    ;
1772 000009BE 6650                      push  ax    ; Exception number
1773 000009C0 66B81000                  mov   ax, KDATA
1774 000009C4 8ED8                      mov   ds, ax
1775 000009C6 8EC0                      mov   es, ax
1776 000009C8 6658                      pop   ax
1777                                    ;
1778 000009CA 891D[38190000]            mov   [FaultOffset], ebx
1779                                    ;
1780 000009D0 88C4                      mov   ah, al
1781 000009D2 240F                      and   al, 0Fh
1782 000009D4 3C09                      cmp   al, 9
1783 000009D6 7602                      jna   short h1ok
1784 000009D8 0407                      add   al, 'A'-':'
1785                                h1ok:
1786 000009DA D0EC                      shr   ah, 1
1787 000009DC D0EC                      shr   ah, 1
1788 000009DE D0EC                      shr   ah, 1
1789 000009E0 D0EC                      shr   ah, 1
1790 000009E2 80FC09                    cmp   ah, 9
1791 000009E5 7603                      jna   short h2ok
1792 000009E7 80C407                    add   ah, 'A'-':'
1793                                h2ok:
1794 000009EA 86E0                      xchg  ah, al
1795 000009EC 66053030                  add   ax, '00'
1796 000009F0 66A3[BC0E0000]            mov   word [excnstr], ax
1797                                    ;
1798                                    ; 29/08/2014
1799 000009F6 A1[38190000]              mov   eax, dword [FaultOffset]
1800 000009FB 51                        push  ecx
1801 000009FC 52                        push  edx
1802 000009FD 89E3                      mov   ebx, esp
1803 000009FF B90A000000                mov   ecx, 10          ; divisor to convert
1804                                                  ; binary number to decimal string
1805                                b2d1:
1806 00000A04 31D2                      xor   edx, edx
1807 00000A06 F7F1                      div   ecx
1808 00000A08 6652                      push  dx
1809 00000A0A 39C8                      cmp   eax, ecx
1810 00000A0C 73F6                      jnb   short b2d1
1811 00000A0E BF[C70E0000]              mov   edi, EIPstr ; EIP value
```

```
                                                        ; points to instruction which faults
                                b2d2:
00000A13 0430                           add     al, '0'
00000A15 AA                             stosb           ; write decimal digit to its place
00000A16 39E3                           cmp     ebx, esp
00000A18 7604                           jna     short b2d3
00000A1A 6658                           pop     ax
00000A1C EBF5                           jmp     short b2d2
                                b2d3:
00000A1E B020                           mov     al, 20h         ; space
00000A20 AA                             stosb
00000A21 30C0                           xor     al, al  ; to do it an ASCIIZ string
00000A23 AA                             stosb
                                        ;
00000A24 5A                             pop     edx
00000A25 59                             pop     ecx
                                        ;
00000A26 B44F                           mov     ah, 4Fh     ; red (4) background,
                                                        ; white (F) forecolor
00000A28 BE[AC0E0000]                   mov     esi, exc_msg ; message offset
                                        ;
00000A2D EB16                           jmp     short piemsg
                                        ;
                                          ;add    dword [scr_row], 0A0h
                                          ;mov    edi, [scr_row]
                                          ;
                                        ;call  printk
                                        ;
                                        ;mov   al, 20h ; END OF INTERRUPT COMMAND TO 8259
                                        ;out   20h, al      ; 8259 PORT
                                        ;
                                        ;pop   es
                                        ;pop   ds
                                        ;pop   edi
                                        ;pop   esi
                                        ;pop   eax
                                        ;iret

                                        ; 23/02/2015
                                        ; 20/08/2014
                                ignore_int:
00000A2F 50                             push   eax
00000A30 53                             push   ebx ; 23/02/2015
00000A31 56                             push   esi
00000A32 57                             push   edi
00000A33 1E                             push   ds
00000A34 06                             push   es
                                        ;
00000A35 B810000000                     mov    eax, KDATA
```

```
1861 00000A3A 8ED8                      mov   ds, ax
1862 00000A3C 8EC0                      mov   es, ax
1863                                    ;
1864 00000A3E B467                      mov   ah, 67h      ; brown (6) background,
1865                                                       ; light gray (7) forecolor
1866 00000A40 BE[6A0E0000]              mov   esi, int_msg ; message offset
1867                                piemsg:
1868                                      ; 27/08/2014
1869 00000A45 8105[0A0C0000]A000-         add     dword [scr_row], 0A0h
1870 00000A4D 0000
1871 00000A4F 8B3D[0A0C0000]              mov     edi, [scr_row]
1872                                      ;
1873 00000A55 E8B7FCFFFF                call  printk
1874                                    ; 23/02/2015
1875 00000A5A B020                      mov   al, 20h  ; END OF INTERRUPT COMMAND TO
1876 00000A5C E6A0                      out   0A0h, al ; the 2nd 8259
1877                                    ;
1878                                iiret:
1879                                    ; 22/08/2014
1880 00000A5E B020                      mov   al, 20h ; END OF INTERRUPT COMMAND TO 8259
1881 00000A60 E620                      out   20h, al      ; 8259 PORT
1882                                    ;
1883 00000A62 07                        pop   es
1884 00000A63 1F                        pop   ds
1885 00000A64 5F                        pop   edi
1886 00000A65 5E                        pop   esi
1887 00000A66 5B                        pop   ebx ; 29/08/2014
1888 00000A67 58                        pop   eax
1889 00000A68 CF                        iret
1890
1891                                    ; 22/02/2015 (dsectpm.s)
1892                                    ; 22/08/2014 - 07/09/2014 (unix386.s)
1893                                rtc_int:    ; Real Time Clock Interrupt (IRQ 8)
1894                                    ; 22/08/2014
1895 00000A69 50                        push  eax
1896 00000A6A 53                        push  ebx ; 29/08/2014
1897 00000A6B 56                        push  esi
1898 00000A6C 57                        push  edi
1899 00000A6D 1E                        push  ds
1900 00000A6E 06                        push  es
1901                                    ;
1902 00000A6F B810000000                mov   eax, KDATA
1903 00000A74 8ED8                      mov   ds, ax
1904 00000A76 8EC0                      mov   es, ax
1905                                    ;
1906                                    ; 22/02/2015
1907 00000A78 BFEC800B00                mov   edi, 0B8000h+0A0h+04Ch ; Row 1, Column 38
1908 00000A7D 807F013F                    cmp     byte [edi+1], 3Fh ; cyan (3) Background
1909                                                  ; white (F) forecolor
```

```
1910                                                    ; (display disk sector frame)
1911 00000A81 7505                         jne    short rtc_np
1912                                       ; 25/08/2014
1913 00000A83 E884000000                   call   rtc_p
1914                                   rtc_np:
1915                                       ; 22/02/2015 - dsectpm.s
1916                                       ; [ source: http://wiki.osdev.org/RTC ]
1917                                       ; read status register C to complete procedure
1918                                       ;(it is needed to get a next IRQ 8)
1919 00000A88 B00C                         mov    al, 0Ch ;
1920 00000A8A E670                         out    70h, al ; select register C
1921 00000A8C 90                           nop
1922 00000A8D E471                         in     al, 71h ; just throw away contents
1923                                       ; 22/02/2015
1924 00000A8F B020                         MOV    AL,EOI       ; END OF INTERRUPT
1925 00000A91 E6A0                         OUT    INTB00,AL    ; FOR CONTROLLER #2
1926                                       ;
1927 00000A93 EBC9                         jmp    short iiret
1928
1929                                       ; 22/08/2014
1930                                       ; IBM PC/AT BIOS source code ----- 10/06/85 (bios.asm)
1931                                       ; (INT 1Ah)
1932                                       ;; Linux (v0.12) source code (main.c) by Linus Torvalds (1991)
1933                                   time_of_day:
1934 00000A95 E84C010000                   call   UPD_IPR                  ; WAIT TILL UPDATE NOT IN PROGRESS
1935 00000A9A 726F                           jc     short rtc_retn
1936 00000A9C B000                         mov    al, CMOS_SECONDS
1937 00000A9E E82B010000                   call   CMOS_READ
1938 00000AA3 A2[D80E0000]                 mov    byte [time_seconds], al
1939 00000AA8 B002                         mov    al, CMOS_MINUTES
1940 00000AAA E81F010000                   call   CMOS_READ
1941 00000AAF A2[D90E0000]                 mov    byte [time_minutes], al
1942 00000AB4 B004                         mov    al, CMOS_HOURS
1943 00000AB6 E813010000                   call   CMOS_READ
1944 00000ABB A2[DA0E0000]                   mov    byte [time_hours], al
1945 00000AC0 B006                         mov    al, CMOS_DAY_WEEK
1946 00000AC2 E807010000                   call   CMOS_READ
1947 00000AC7 A2[DB0E0000]                 mov    byte [date_wday], al
1948 00000ACC B007                         mov    al, CMOS_DAY_MONTH
1949 00000ACE E8FB000000                   call   CMOS_READ
1950 00000AD3 A2[DC0E0000]                 mov    byte [date_day], al
1951 00000AD8 B008                         mov    al, CMOS_MONTH
1952 00000ADA E8EF000000                   call   CMOS_READ
1953 00000ADF A2[DD0E0000]                 mov    byte [date_month], al
1954 00000AE4 B009                         mov    al, CMOS_YEAR
1955 00000AE6 E8E3000000                   call   CMOS_READ
1956 00000AEB A2[DE0E0000]                 mov    byte [date_year], al
1957 00000AF0 B032                         mov    al, CMOS_CENTURY
1958 00000AF2 E8D7000000                   call   CMOS_READ
```

```
1959 00000AF7 A2[DF0E0000]              mov   byte [date_century], al
1960                                    ;
1961 00000AFC B000                      mov   al, CMOS_SECONDS
1962 00000AFE E8CB000000                call  CMOS_READ
1963 00000B03 3A05[D80E0000]            cmp   al, byte [time_seconds]
1964 00000B09 758A                      jne   short time_of_day
1965
1966                                rtc_retn:
1967 00000B0B C3                        retn
1968
1969                                rtc_p:
1970                                    ; 22/02/2015 dsectpm.s
1971                                    ; 25/08/2014 - 07/09/2014 unix386.s
1972                                    ; Print Real Time Clock content
1973                                    ;
1974 00000B0C E884FFFFFF                call  time_of_day
1975 00000B11 72F8                      jc    short rtc_retn
1976                                    ;
1977 00000B13 3A05[2B0F0000]            cmp   al, byte [ptime_seconds]
1978 00000B19 74F0                        je      short rtc_retn ; 29/08/2014
1979                                    ;
1980 00000B1B A2[2B0F0000]              mov   byte [ptime_seconds], al
1981                                    ;
1982 00000B20 A0[DF0E0000]              mov   al, byte [date_century]
1983 00000B25 E8D7000000                call  bcd_to_ascii
1984 00000B2A 66A3[F80E0000]            mov   word [datestr+6], ax
1985 00000B30 A0[DE0E0000]              mov   al, byte [date_year]
1986 00000B35 E8C7000000                call  bcd_to_ascii
1987 00000B3A 66A3[FA0E0000]            mov   word [datestr+8], ax
1988 00000B40 A0[DD0E0000]              mov   al, byte [date_month]
1989 00000B45 E8B7000000                call  bcd_to_ascii
1990 00000B4A 66A3[F50E0000]            mov   word [datestr+3], ax
1991 00000B50 A0[DC0E0000]              mov   al, byte [date_day]
1992 00000B55 E8A7000000                call  bcd_to_ascii
1993 00000B5A 66A3[F20E0000]            mov   word [datestr], ax
1994                                    ;
1995 00000B60 31DB                      xor   ebx, ebx
1996 00000B62 8A1D[DB0E0000]            mov   bl, byte [date_wday]
1997 00000B68 C0E302                    shl   bl, 2
1998 00000B6B 81C3[0B0F0000]            add   ebx, daytmp
1999 00000B71 8B03                      mov   eax, dword [ebx]
2000 00000B73 A3[FD0E0000]              mov   dword [daystr], eax
2001                                    ;
2002 00000B78 A0[DA0E0000]              mov   al, byte [time_hours]
2003 00000B7D E87F000000                call  bcd_to_ascii
2004 00000B82 66A3[010F0000]            mov   word [timestr], ax
2005 00000B88 A0[D90E0000]              mov   al, byte [time_minutes]
2006 00000B8D E86F000000                call  bcd_to_ascii
2007 00000B92 66A3[040F0000]            mov   word [timestr+3], ax
```

```
2008 00000B98 A0[D80E0000]              mov   al, byte [time_seconds]
2009 00000B9D E85F000000                call  bcd_to_ascii
2010 00000BA2 66A3[070F0000]            mov   word [timestr+6], ax
2011                                    ;
2012 00000BA8 BE[E00E0000]              mov   esi, rtc_msg ; message offset
2013                                    ; 22/02/2015
2014                                    ;mov  edi, 0B8000h+0A0h+04Ch ; Row 1, Column 38
2015                                    ;mov  ah, [edi+1]
2016                                    ;cmp  ah, 3Fh ; cyan (3) Background
2017                                          ; white (F) forecolor
2018                                          ; (display disk sector frame)
2019                                    ;jne  short prtcmsg_ok
2020                                prtcmsg:
2021 00000BAD AC                        lodsb
2022 00000BAE 08C0                      or    al, al
2023 00000BB0 7404                      jz    short prtcmsg_ok
2024 00000BB2 AA                        stosb
2025 00000BB3 47                        inc   edi
2026 00000BB4 EBF7                      jmp   short prtcmsg
2027                                prtcmsg_ok:
2028 00000BB6 C3                        retn
2029
2030                                ; Default IRQ 7 handler against spurious IRQs (from master PIC)
2031                                ; 25/02/2015 (source: http://wiki.osdev.org/8259_PIC)
2032                                default_irq7:
2033 00000BB7 6650                      push  ax
2034 00000BB9 B00B                      mov   al, 0Bh  ; In-Service register
2035 00000BBB E620                      out   20h, al
2036 00000BBD EB00                        jmp short $+2
2037 00000BBF EB00                      jmp short $+2
2038 00000BC1 E420                      in    al, 20h
2039 00000BC3 2480                      and   al, 80h ; bit 7 (is it real IRQ 7 or fake?)
2040 00000BC5 7404                        jz      short irq7_iret ; Fake (spurious) IRQ, do not send EOI
2041 00000BC7 B020                        mov     al, 20h ; EOI
2042 00000BC9 E620                      out   20h, al
2043                                irq7_iret:
2044 00000BCB 6658                      pop   ax
2045 00000BCD CF                        iretd
2046
2047                                    ; 22/08/2014
2048                                    ; IBM PC/AT BIOS source code ----- 10/06/85 (test4.asm)
2049                                CMOS_READ:
2050 00000BCE 9C                        pushf         ; SAVE INTERRUPT ENABLE STATUS AND FLAGS
2051 00000BCF D0C0                      rol   al, 1  ; MOVE NMI BIT TO LOW POSITION
2052 00000BD1 F9                        stc           ; FORCE NMI BIT ON IN CARRY FLAG
2053 00000BD2 D0D8                      rcr   al, 1  ; HIGH BIT ON TO DISABLE NMI - OLD IN CY
2054 00000BD4 FA                        cli           ; DISABLE INTERRUPTS
2055 00000BD5 E670                      out   CMOS_PORT, al; ADDRESS LOCATION AND DISABLE NMI
2056 00000BD7 90                        nop           ; I/O DELAY
```

```
2057 00000BD8 E471                      in    al, CMOS_DATA; READ THE REQUESTED CMOS LOCATION
2058 00000BDA 6650                      push  ax    ; SAVE (AH) REGISTER VALUE AND CMOS BYTE
2059 00000BDC B01A                      mov   al, CMOS_REG_D*2 ; GET ADDRESS OF DEFAULT LOCATION
2060 00000BDE D0D8                      rcr   al, 1  ; PUT ORIGINAL NMI MASK BIT INTO ADDRESS
2061 00000BE0 E670                      out   CMOS_PORT, al; SET DEFAULT TO READ ONLY REGISTER
2062 00000BE2 6658                      pop   ax    ; RESTORE (AH) AND (AL), CMOS BYTE
2063 00000BE4 9D                        popf
2064 00000BE5 C3                        retn        ; RETURN WITH FLAGS RESTORED
2065
2066                                    ; 22/08/2014
2067                                    ; IBM PC/AT BIOS source code ----- 10/06/85 (bios2.asm)
2068                                UPD_IPR:                    ; WAIT TILL UPDATE NOT IN PROGRESS
2069 00000BE6 51                        push  ecx
2070 00000BE7 B9FFFF0000                mov   ecx, 65535        ; SET TIMEOUT LOOP COUNT (= 800)
2071                                          ; mov cx, 800
2072                                UPD_10:
2073 00000BEC B00A                      mov   al, CMOS_REG_A          ; ADDRESS STATUS REGISTER A
2074 00000BEE FA                        cli                     ; NO TIMER INTERRUPTS DURING UPDATES
2075 00000BEF E8DAFFFFFF                call  CMOS_READ         ; READ UPDATE IN PROCESS FLAG
2076 00000BF4 A880                      test  al, 80h                ; IF UIP BIT IS ON ( CANNOT READ TIME )
2077 00000BF6 7406                      jz    short UPD_90      ; EXIT WITH CY= 0 IF CAN READ CLOCK NOW
2078 00000BF8 FB                        sti                     ; ALLOW INTERRUPTS WHILE WAITING
2079 00000BF9 E2F1                      loop  UPD_10            ; LOOP TILL READY OR TIMEOUT
2080 00000BFB 31C0                      xor   eax, eax          ; CLEAR RESULTS IF ERROR
2081                                          ; xor ax, ax
2082 00000BFD F9                        stc                     ; SET CARRY FOR ERROR
2083                                UPD_90:
2084 00000BFE 59                        pop   ecx               ; RESTORE CALLERS REGISTER
2085 00000BFF FA                        cli                     ; INTERRUPTS OFF DURING SET
2086 00000C00 C3                        retn                    ; RETURN WITH CY FLAG SET
2087
2088                                bcd_to_ascii:
2089                                    ; 25/08/2014
2090                                    ; INPUT ->
2091                                    ;     al = Packed BCD number
2092                                    ; OUTPUT ->
2093                                    ;     ax = ASCII word/number
2094                                    ;
2095                                    ; Erdogan Tan - 1998 (proc_hex) - TRDOS.ASM (2004-2011)
2096                                    ;
2097 00000C01 D410                     db 0D4h,10h                  ; Undocumented inst. AAM
2098                                                            ; AH = AL / 10h
2099                                                            ; AL = AL MOD 10h
2100 00000C03 660D3030                 or ax,'00'                   ; Make it ASCII based
2101
2102 00000C07 86E0                       xchg ah, al
2103
2104 00000C09 C3                       retn
2105
```

```
                                ; 27/08/2014
                                scr_row:
00000C0A E0810B00                     dd 0B8000h + 0A0h + 0A0h + 0A0h ; Row 3
                                scr_col:
00000C0E 00000000                     dd 0

00000C12 90<rept>               ALIGN 8

                                gdt:  ; Global Descriptor Table (29/12/2013)
00000C18 0000000000000000             dw 0, 0, 0, 0       ; NULL descriptor
                                      ; 18/08/2014
                                                  ; 8h kernel code segment, base = 00000000h
00000C20 FFFF0000009A                 dw 0FFFFh, 0, 9A00h
00000C26 C300                         dw 00C3h            ; KCODE
                                                  ; 10h kernel data segment, base = 00000000h
00000C28 FFFF00000092                 dw 0FFFFh, 0, 9200h
00000C2E C300                         dw 00C3h            ; KDATA
                                                  ; 1Bh user code segment, base address = 0
                                      ;dw 0FFFFh, 0, 0FA00h, 0CBh ; UCODE
                                                  ; 23h user data segment, base address = 0
                                      ;dw 0FFFFh, 0, 0F200h, 0CBh ; UDATA

                                gdt_end:
                                      ;; 9Ah = 1001 1010b (GDT byte 5) P=1/DPL=00/1/TYPE=1010,
                                                        ;; Type= 1 (code)/C=0/R=1/A=0
                                            ; P= Present, DPL=0=ring 0,  1= user (0= system)
                                            ; 1= Code C= non-Conforming, R= Readable, A = Accessed

                                      ;; 92h = 1001 0010b (GDT byte 5) P=1/DPL=00/1/TYPE=1010,
                                                        ;; Type= 0 (data)/E=0/W=1/A=0
                                            ; P= Present, DPL=0=ring 0,  1= user (0= system)
                                            ; 0= Data E= Expansion direction (1= down, 0= up)
                                            ; W= Writable, A= Accessed

                                      ;; FAh = 1111 1010b (GDT byte 5) P=1/DPL=11/1/TYPE=1010,
                                                        ;; Type= 1 (code)/C=0/R=1/A=0
                                            ; P= Present, DPL=3=ring 3,  1= user (0= system)
                                            ; 1= Code C= non-Conforming, R= Readable, A = Accessed

                                      ;; F2h = 1111 0010b (GDT byte 5) P=1/DPL=11/1/TYPE=1010,
                                                        ;; Type= 0 (data)/E=0/W=1/A=0
                                            ; P= Present, DPL=3=ring 3,  1= user (0= system)
                                            ; 0= Data E= Expansion direction (1= down, 0= up)

                                      ;; C3h = 1100 0011b (GDT byte 6) G=1/B=1/0/AVL=0, Limit=0011b (3)
                                            ;; Limit = 3FFFFh (=> 3FFFFh+1= 40000h)
                                            ;       = 40000h * 1000h (G=1) = 1GB
                                            ; G= Granularity (1= 4KB), B=Big (32 bit),
                                            ; AVL= Available to programmers
```

```
2155                                        ;; CBh = 1100 1011b (GDT byte 6) G=1/DB=1/0/AVL=0, Limit=1011b (3)
2156                                               ;; Limit = BFFFFh (=> BFFFFh+1= C0000h)
2157                                               ;       = C0000h * 1000h (G=1) = 3GB
2158                                               ; G= Granularity (1= 4KB),
2159                                               ; D=Default operand size (32 bit),
2160                                               ; AVL= Available to programmers
2161
2162
2163                                        ; Interrupt Descriptor Table (20/08/2014)
2164                                  idt:
2165 00000C30 00<rept>                        times  64*8 db 0   ; INT 0 to INT 3Fh
2166                                  idt_end:
2167
2168
2169                                  gdtd:
2170 00000E30 1700                              dw gdt_end - gdt - 1    ; Limit (size)
2171 00000E32 [180C0000]                        dd gdt                  ; Address of the GDT
2172
2173                                        ; 20/08/2014
2174                                  idtd:
2175 00000E36 FF01                              dw idt_end - idt - 1    ; Limit (size)
2176 00000E38 [300C0000]                        dd idt                  ; Address of the IDT
2177
2178
2179
2180 00000E3C 00                      db 0
2181 00000E3D 90                      Align 2
2182
2183                                  msgPM:
2184 00000E3E 50726F746563746564-               db "Protected mode and paging are ENABLED ... ", 0
2185 00000E47 206D6F646520616E64-
2186 00000E50 20706167696E672061-
2187 00000E59 726520454E41424C45-
2188 00000E62 44202E2E2E2000
2189
2190 00000E69 90                      Align 2
2191
2192                                  ; 20/08/2014
2193                                    ; /* This is the default interrupt "handler" :-) */
2194                                    ; Linux v0.12 (head.s)
2195                                  int_msg:
2196 00000E6A 556E6B6E6F776E2069-           db "Unknown interrupt ! ", 0
2197 00000E73 6E7465727275707420-
2198 00000E7C 212000
2199
2200 00000E7F 90                      Align 2
2201
2202                                  ; 21/08/2014
2203                                  timer_msg:
```

```
2204 00000E80 49525120302028494E-          db "IRQ 0 (INT 20h) ! Timer Interrupt : "
2205 00000E89 542032306829202120-
2206 00000E92 54696D657220496E74-
2207 00000E9B 657272757074203A20
2208                               tcountstr:
2209 00000EA4 303030303020                 db "00000 "
2210 00000EAA 00                           db 0
2211
2212 00000EAB 90                   Align 2
2213                                       ; 21/08/2014
2214                               exc_msg:
2215 00000EAC 435055206578636570-          db "CPU exception ! "
2216 00000EB5 74696F6E202120
2217                               excnstr:           ; 25/08/2014
2218 00000EBC 3F3F68202045495020-          db "??h", "  EIP : "
2219 00000EC5 3A20
2220                                       ; 29/08/2014
2221 00000EC7 00<rept>             EIPstr: times 12 db 0
2222
2223
2224 00000ED3 90                   Align 4
2225
2226                               tcount:
2227 00000ED4 00000000                     dd 0
2228
2229                               ; 22/08/2014 (RTC)
2230                               ; (Packed BCD)
2231 00000ED8 00                   time_seconds: db 0
2232 00000ED9 00                   time_minutes: db 0
2233 00000EDA 00                   time_hours:   db 0
2234 00000EDB 00                   date_wday:    db 0
2235 00000EDC 00                   date_day:     db 0
2236 00000EDD 00                   date_month:   db 0
2237 00000EDE 00                   date_year:    db 0
2238 00000EDF 00                   date_century: db 0
2239
2240                               rtc_msg:
2241 00000EE0 5265616C2054696D65-          db "Real Time Clock - "
2242 00000EE9 20436C6F636B202D20
2243                               datestr:
2244 00000EF2 30302F30302F303030-          db "00/00/0000"
2245 00000EFB 30
2246 00000EFC 20                           db " "
2247                               daystr:
2248 00000EFD 44415920                     db "DAY "
2249                               timestr:
2250 00000F01 30303A30303A3030               db "00:00:00"
2251 00000F09 20                           db " "
2252 00000F0A 00                           db 0
```

```
                                daytmp:
                                      ; 28/02/2015
00000F0B 3F3F3F2053554E204D-          db "??? SUN MON TUE WED THU FRI SAT "
00000F14 4F4E20545545205745-
00000F1D 442054485520465249-
00000F26 2053415420

00000F2B FF                     ptime_seconds: db 0FFh

                                      ;;23/02/2015
                                      ; 25/08/2014
                                ;scounter:
                                ;     db 5
                                ;     db 19

                                ; 20/02/2015
                                ; 03/12/2014
                                ; 07/09/2014
                                ; KEYBOARD INTERRUPT HANDLER
                                ; (kb_int - Retro UNIX 8086 v1 - U0.ASM, 30/06/2014)

                                keyb_int:
                                      ; 25/02/2015
                                      ; 20/02/2015
                                      ; 03/12/2014 (getc_int - INT 16h modifications)
                                      ; 07/09/2014 - Retro UNIX 386 v1
                                      ; 30/06/2014
                                      ; 10/05/2013
                                            ; Retro Unix 8086 v1 feature only!
                                      ; 03/03/2014

00000F2C 1E                           push  ds
00000F2D 53                           push  ebx
00000F2E 50                           push  eax
                                      ;
00000F2F 66B81000                     mov   ax, KDATA
00000F33 8ED8                         mov   ds, ax
                                      ;
00000F35 9C                           pushfd
00000F36 0E                           push  cs
00000F37 E818020000                   call  kb_int
                                      ;
00000F3C B411                         mov   ah, 11h       ; 03/12/2014
00000F3E E84B000000                   call  getc
00000F43 7445                         jz    short keyb_int4
                                      ;
00000F45 B410                         mov   ah, 10h       ; 03/12/2014
00000F47 E842000000                   call  getc
                                      ;
```

```
                                        ; 20/02/2015
00000F4C 31DB                           xor   ebx, ebx
00000F4E 8A1D[4C490000]                 mov   bl, [ptty]  ; active_page
                                        ;
00000F54 20C0                           and   al, al
00000F56 7527                           jnz   short keyb_int1
                                        ;
00000F58 80FC68                         cmp   ah, 68h      ; ALT + F1 key
00000F5B 7222                           jb    short keyb_int1
00000F5D 80FC6F                         cmp   ah, 6Fh  ; ALT + F8 key
00000F60 771D                           ja    short keyb_int1
                                        ;
00000F62 88DF                           mov   bh, bl
00000F64 80C768                         add   bh, 68h
00000F67 38E7                           cmp   bh, ah
00000F69 740C                           je    short keyb_int0
00000F6B 88E0                           mov   al, ah
00000F6D 2C68                           sub   al, 68h
00000F6F E8D10C0000                     call  tty_sw
00000F74 6631C0                         xor   ax, ax
                                keyb_int0: ; 25/02/2015
00000F77 31DB                           xor   ebx, ebx
00000F79 8A1D[4C490000]                 mov   bl, [ptty]  ; active_page
                                keyb_int1:
00000F7F D0E3                           shl   bl, 1
00000F81 81C3[52490000]                 add   ebx, ttychr
                                        ;
                                        ;or   ax, ax
                                        ;jz   short keyb_int2
                                        ;
                                        ;cmp  word [ebx], 0
                                              ;ja   short kb_int_3
                                keyb_int2:
00000F87 668903                           mov  word [ebx], ax  ; Save ascii code
                                                               ; and scan code of the character
                                                               ; for current tty (or last tty
                                                               ; just before tty switch).
                                keyb_int3:
                                          ;mov     al, byte [ptty]
                                        ;call  wakeup
                                        ;
                                keyb_int4:
00000F8A 58                             pop   eax
00000F8B 5B                             pop   ebx
00000F8C 1F                             pop   ds
00000F8D CF                             iret

                                ; 18/02/2015
                                ; REMINDER: Only 'keyb_int' (IRQ 9) must call getc.
```

```
                                ; 'keyb_int' always handles 'getc' at 1st and puts the
                                ; scancode and ascii code of the character
                                ; in the tty input (ttychr) buffer.
                                ; Test procedures must call 'getch' for tty input
                                ; otherwise, 'getc' will not be able to return to the caller
                                ; due to infinite (key press) waiting loop.
                                ;
                                ; 03/12/2014
                                ; 26/08/2014
                                ; KEYBOARD I/O
                                ; (INT_16h - Retro UNIX 8086 v1 - U9.ASM, 30/06/2014)

                                ;NOTE: 'k0' to 'k7' are name of OPMASK registers.
                                ;     (The reason of using '_k' labels!!!) (27/08/2014)
                                ;NOTE: 'NOT' keyword is '~' unary operator in NASM.
                                ;     ('NOT LC_HC' --> '~LC_HC') (bit reversing operator)

                                getc:
00000F8E 9C                           pushfd ; 28/08/2014
00000F8F 0E                           push   cs
00000F90 E801000000                   call   getc_int
00000F95 C3                           retn

                                getc_int:
                                      ; 28/02/2015
                                      ; 03/12/2014 (derivation from pc-xt-286 bios source code -1986-,
                                      ;            instead of pc-at bios - 1985-)
                                      ; 28/08/2014 (_k1d)
                                      ; 30/06/2014
                                      ; 03/03/2014
                                      ; 28/02/2014
                                      ; Derived from "KEYBOARD_IO_1" procedure of IBM "pc-xt-286"
                                      ; rombios source code (21/04/1986)
                                      ;       'keybd.asm', INT 16H, KEYBOARD_IO
                                      ;
                                      ; KYBD --- 03/06/86  KEYBOARD BIOS
                                      ;
                                      ;--- INT 16 H ----------------------------------------------------------------
-
                                      ; KEYBOARD I/O                                                          :
                                      ;     THESE ROUTINES PROVIDE READ KEYBOARD SUPPORT                      :
                                      ; INPUT                                                                :
                                      ;     (AH)= 00H  READ THE NEXT ASCII CHARACTER ENTERED FROM THE KEYBOARD,   :
                                      ;                RETURN THE RESULT IN (AL), SCAN CODE IN (AH).             :
                                      ;                THIS IS THE COMPATIBLE READ INTERFACE, EQUIVALENT TO THE   :
                                      ;                   STANDARD PC OR PCAT KEYBOARD
:
                                      ;------------------------------------------------------------------------
:
```

```
;     (AH)= 01H  SET THE ZERO FLAG TO INDICATE IF AN ASCII CHARACTER IS     :
;              AVAILABLE TO BE READ FROM THE KEYBOARD BUFFER.          :
;              (ZF)= 1 -- NO CODE AVAILABLE                             :
;              (ZF)= 0 -- CODE IS AVAILABLE  (AX)= CHARACTER              :
;              IF (ZF)= 0, THE NEXT CHARACTER IN THE BUFFER TO BE READ IS :
;              IN (AX), AND THE ENTRY REMAINS IN THE BUFFER.              :
;              THIS WILL RETURN ONLY PC/PCAT KEYBOARD COMPATIBLE CODES    :
;-------------------------------------------------------------------------
:
;     (AH)= 02H  RETURN THE CURRENT SHIFT STATUS IN AL REGISTER             :
;              THE BIT SETTINGS FOR THIS CODE ARE INDICATED IN THE        :
;              EQUATES FOR @KB_FLAG                                        :
;-------------------------------------------------------------------------
:
;     (AH)= 03H  SET TYPAMATIC RATE AND DELAY                               :
;           (AL) = 05H                                                      :
;           (BL) = TYPAMATIC RATE (BITS 5 - 7 MUST BE RESET TO 0)           :
;                                                                          :
;                    REGISTER     RATE      REGISTER     RATE
:
;                     VALUE     SELECTED     VALUE     SELECTED
:
;                    ------------------------------------------
:
;                   00H        30.0        10H        7.5                 :
;                   01H        26.7        11H        6.7                 :
;                   02H        24.0        12H        6.0                 :
;                   03H        21.8        13H        5.5                 :
;                   04H        20.0        14H        5.0                 :
;                   05H        18.5        15H        4.6                 :
;                   06H        17.1        16H        4.3                 :
;                   07H        16.0        17H        4.0                 :
;                   08H        15.0        18H        3.7                 :
;                   09H        13.3        19H        3.3                 :
;                   0AH        12.0        1AH        3.0                 :
;                   0BH        10.9        1BH        2.7                 :
;                   0CH        10.0        1CH        2.5                 :
;                   0DH         9.2        1DH        2.3                 :
;                   0EH         8.6        1EH        2.1                 :
;                   0FH         8.0        1FH        2.0                 :
;                                                                        :
;           (BH) = TYPAMATIC DELAY  (BITS 2 - 7 MUST BE RESET TO 0)         :
;                                                                          :
;                    REGISTER     DELAY
:
;                     VALUE       VALUE
:
;                    -----------------
:
```

```
                        ;                   00H        250 ms                                    :
                        ;                   01H        500 ms                                    :
                        ;                   02H        750 ms                                    :
                        ;                   03H       1000 ms                                    :
                        ;-------------------------------------------------------------------------
:
                        ;     (AH)= 05H  PLACE ASCII CHARACTER/SCAN CODE COMBINATION IN KEYBOARD    :
                        ;               BUFFER AS IF STRUCK FROM KEYBOARD                         :
                        ;               ENTRY:  (CL) = ASCII CHARACTER                          :
                        ;                       (CH) = SCAN CODE                                  :
                        ;               EXIT:   (AH) = 00H = SUCCESSFUL OPERATION                 :
                        ;                       (AL) = 01H = UNSUCCESSFUL - BUFFER FULL           :
                        ;               FLAGS:  CARRY IF ERROR                                    :
                        ;-------------------------------------------------------------------------
:
                        ;     (AH)= 10H  EXTENDED READ INTERFACE FOR THE ENHANCED KEYBOARD,         :
                        ;               OTHERWISE SAME AS FUNCTION AH=0                           :
                        ;-------------------------------------------------------------------------
:
                        ;     (AH)= 11H  EXTENDED ASCII STATUS FOR THE ENHANCED KEYBOARD,           :
                        ;               OTHERWISE SAME AS FUNCTION AH=1                           :
                        ;-------------------------------------------------------------------------
:
                        ;     (AH)= 12H  RETURN THE EXTENDED SHIFT STATUS IN AX REGISTER            :
                        ;               AL = BITS FROM KB_FLAG, AH = BITS FOR LEFT AND RIGHT      :
                        ;               CTL AND ALT KEYS FROM KB_FLAG_1 AND KB_FLAG_3             :
                        ; OUTPUT                                                                 :
                        ;     AS NOTED ABOVE, ONLY (AX) AND FLAGS CHANGED                        :
                        ;     ALL REGISTERS RETAINED                                              :
                        ;-------------------------------------------------------------------------
-

00000F96 FB             sti                     ; INTERRUPTS BACK ON
00000F97 1E             push   ds               ; SAVE CURRENT DS
00000F98 53             push   ebx              ; SAVE BX TEMPORARILY
                        ;push  ecx              ; SAVE CX TEMPORARILY
00000F99 66BB1000         mov     bx, KDATA
00000F9D 8EDB           mov    ds, bx           ; PUT SEGMENT VALUE OF DATA AREA INTO DS
00000F9F 08E4           or     ah, ah           ; CHECK FOR (AH)= 00H
00000FA1 7439           jz     short _K1        ; ASCII_READ
00000FA3 FECC           dec    ah                  ; CHECK FOR (AH)= 01H
00000FA5 7452             jz      short _K2            ; ASCII_STATUS
00000FA7 FECC           dec    ah               ; CHECK FOR (AH)= 02H
00000FA9 0F8485000000     jz      _K3                  ; SHIFT STATUS
00000FAF FECC           dec    ah               ; CHECK FOR (AH)= 03H
00000FB1 0F8484000000     jz      _K300                ; SET TYPAMATIC RATE/DELAY
00000FB7 80EC02         sub    ah, 2            ; CHECK FOR (AH)= 05H
00000FBA 0F84A1000000     jz      _K500                ; KEYBOARD WRITE
                      _KIO1:
```

```
2482 00000FC0 80EC0B                    sub    ah, 11              ; AH =  10H
2483 00000FC3 740B                      jz     short _K1E          ; EXTENDED ASCII READ
2484 00000FC5 FECC                      dec    ah                  ; CHECK FOR (AH)= 11H
2485 00000FC7 7421                      jz     short _K2E          ; EXTENDED_ASCII_STATUS
2486 00000FC9 FECC                      dec    ah                  ; CHECK FOR (AH)= 12H
2487 00000FCB 7449                      jz     short _K3E          ; EXTENDED_SHIFT_STATUS
2488                                _KIO_EXIT:
2489                                    ;pop   ecx                 ; RECOVER REGISTER
2490 00000FCD 5B                        pop    ebx                 ; RECOVER REGISTER
2491 00000FCE 1F                        pop    ds                  ; RECOVER SEGMENT
2492 00000FCF CF                        iretd                      ; INVALID COMMAND, EXIT
2493
2494                                    ;-----ASCII CHARACTER
2495                                _K1E:
2496 00000FD0 E8B9000000                call   _K1S                ; GET A CHARACTER FROM THE BUFFER (EXTENDED)
2497 00000FD5 E82E010000                call   _KIO_E_XLAT         ; ROUTINE TO XLATE FOR EXTENDED CALLS
2498 00000FDA EBF1                      jmp    short _KIO_EXIT        ; GIVE IT TO THE CALLER
2499                                _K1:
2500 00000FDC E8AD000000                call   _K1S                ; GET A CHARACTER FROM THE BUFFER
2501 00000FE1 E82D010000                call   _KIO_S_XLAT         ; ROUTINE TO XLATE FOR STANDARD CALLS
2502 00000FE6 72F4                      jc     short _K1           ; CARRY SET MEANS TROW CODE AWAY
2503                                _K1A:
2504 00000FE8 EBE3                      jmp    short _KIO_EXIT         ; RETURN TO CALLER
2505
2506                                    ;-----ASCII STATUS
2507                                _K2E:
2508 00000FEA E8EA000000                call   _K2S                ; TEST FOR CHARACTER IN BUFFER (EXTENDED)
2509 00000FEF 7420                      jz     short _K2B          ; RETURN IF BUFFER EMPTY
2510 00000FF1 9C                        pushf                      ; SAVE ZF FROM TEST
2511 00000FF2 E811010000                call   _KIO_E_XLAT         ; ROUTINE TO XLATE FOR EXTENDED CALLS
2512 00000FF7 EB17                      jmp    short _K2A           ; GIVE IT TO THE CALLER
2513                                _K2:
2514 00000FF9 E8DB000000                call   _K2S                ; TEST FOR CHARACTER IN BUFFER
2515 00000FFE 7411                      jz     short _K2B          ; RETURN IF BUFFER EMPTY
2516 00001000 9C                        pushf                      ; SAVE ZF FROM TEST
2517 00001001 E80D010000                call   _KIO_S_XLAT         ; ROUTINE TO XLATE FOR STANDARD CALLS
2518 00001006 7308                      jnc    short _K2A           ; CARRY CLEAR MEANS PASS VALID CODE
2519 00001008 9D                        popf                       ; INVALID CODE FOR THIS TYPE OF CALL
2520 00001009 E880000000                call   _K1S                ; THROW THE CHARACTER AWAY
2521 0000100E EBE9                      jmp    short _K2           ; GO LOOK FOR NEXT CHAR, IF ANY
2522                                _K2A:
2523 00001010 9D                        popf                       ; RESTORE ZF FROM TEST
2524                                _K2B:
2525                                    ;pop   ecx                 ; RECOVER REGISTER
2526 00001011 5B                        pop    ebx                 ; RECOVER REGISTER
2527 00001012 1F                        pop    ds                  ; RECOVER SEGMENT
2528 00001013 CA0400                    retf   4                   ; THROW AWAY (e)FLAGS
2529
2530                                    ;-----SHIFT STATUS
```

```
2531                                    _K3E:                                    ; GET THE EXTENDED SHIFT STATUS FLAGS
2532 00001016 8A25[00190000]                    mov    ah, [KB_FLAG_1]           ; GET SYSTEM SHIFT KEY STATUS
2533 0000101C 80E404                            and    ah, SYS_SHIFT       ; MASK ALL BUT SYS KEY BIT
2534                                            ;mov   cl, 5               ; SHIFT THEW SYSTEMKEY BIT OVER TO
2535                                            ;shl   ah, cl              ; BIT 7 POSITION
2536 0000101F C0E405                              shl  ah, 5
2537 00001022 A0[00190000]                      mov    al, [KB_FLAG_1]           ; GET SYSTEM SHIFT STATES BACK
2538 00001027 2473                              and    al, 01110011b       ; ELIMINATE SYS SHIFT, HOLD_STATE AND INS_SHIFT
2539 00001029 08C4                              or     ah, al                 ; MERGE REMAINING BITS INTO AH
2540 0000102B A0[02190000]                      mov    al, [KB_FLAG_3]           ; GET RIGHT CTL AND ALT
2541 00001030 240C                              and    al, 00001100b       ; ELIMINATE LC_E0 AND LC_E1
2542 00001032 08C4                              or     ah, al              ; OR THE SHIFT FLAGS TOGETHER
2543                                    _K3:
2544 00001034 A0[FF180000]                      mov    al, [KB_FLAG]       ; GET THE SHIFT STATUS FLAGS
2545 00001039 EB92                              jmp    short _KIO_EXIT           ; RETURN TO CALLER
2546
2547                                            ;----- SET TYPAMATIC RATE AND DELAY
2548                                    _K300:
2549 0000103B 3C05                              cmp    al, 5               ; CORRECT FUNCTION CALL?
2550 0000103D 758E                              jne    short _KIO_EXIT           ; NO, RETURN
2551 0000103F F6C3E0                            test   bl, 0E0h            ; TEST FOR OUT-OF-RANGE RATE
2552 00001042 7589                              jnz    short _KIO_EXIT           ; RETURN IF SO
2553 00001044 F6C7FC                            test   BH, 0FCh            ; TEST FOR OUT-OF-RANGE DELAY
2554 00001047 7584                              jnz    short _KIO_EXIT           ; RETURN IF SO
2555 00001049 B0F3                              mov    al, KB_TYPA_RD            ; COMMAND FOR TYPAMATIC RATE/DELAY

2556 0000104B E8A8060000                        call   SND_DATA           ; SEND TO KEYBOARD
2557                                            ;mov   cx, 5               ; SHIFT COUNT
2558                                            ;shl   bh, cl              ; SHIFT DELAY OVER
2559 00001050 C0E705                            shl    bh, 5
2560 00001053 88D8                              mov    al, bl              ; PUT IN RATE
2561 00001055 08F8                              or     al, bh              ; AND DELAY
2562 00001057 E89C060000                        call   SND_DATA           ; SEND TO KEYBOARD
2563 0000105C E96CFFFFFF                          jmp    _KIO_EXIT               ; RETURN TO CALLER
2564
2565                                            ;----- WRITE TO KEYBOARD BUFFER
2566                                    _K500:
2567 00001061 56                                push   esi                 ; SAVE SI (esi)
2568 00001062 FA                                cli                        ;
2569 00001063 8B1D[10190000]                    mov    ebx, [BUFFER_TAIL]  ; GET THE 'IN TO' POINTER TO THE BUFFER
2570 00001069 89DE                              mov    esi, ebx            ; SAVE A COPY IN CASE BUFFER NOT FULL
2571 0000106B E8D3000000                        call   _K4                 ; BUMP THE POINTER TO SEE IF BUFFER IS FULL
2572 00001070 3B1D[0C190000]                    cmp    ebx, [BUFFER_HEAD]  ; WILL THE BUFFER OVERRUN IF WE STORE THIS?
2573 00001076 740D                              je     short _K502         ; YES - INFORM CALLER OF ERROR
2574 00001078 66890E                            mov    [esi], cx           ; NO - PUT ASCII/SCAN CODE INTO BUFFER
2575 0000107B 891D[10190000]                    mov    [BUFFER_TAIL], ebx  ; ADJUST 'IN TO' POINTER TO REFLECT CHANGE
2576 00001081 28C0                              sub    al, al              ; TELL CALLER THAT OPERATION WAS SUCCESSFUL
2577 00001083 EB02                              jmp    short _K504         ; SUB INSTRUCTION ALSO RESETS CARRY FLAG
2578                                    _K502:
```

```
2579 00001085 B001                      mov   al, 01h                   ; BUFFER FULL INDICATION
2580                                  _K504:
2581 00001087 FB                        sti
2582 00001088 5E                        pop   esi                ; RECOVER SI (esi)
2583 00001089 E93FFFFFFF                  jmp     _KIO_EXIT               ; RETURN TO CALLER WITH STATUS IN AL
2584
2585                                    ;----- READ THE KEY TO FIGURE OUT WHAT TO DO -----
2586                                  _K1S:
2587 0000108E FA                        cli   ; 03/12/2014
2588 0000108F 8B1D[0C190000]              mov     ebx, [BUFFER_HEAD]     ; GET POINTER TO HEAD OF BUFFER
2589 00001095 3B1D[10190000]              cmp     ebx, [BUFFER_TAIL]     ; TEST END OF BUFFER
2590                                    ;jne   short _K1U          ; IF ANYTHING IN BUFFER SKIP INTERRUPT
2591 0000109B 750F                      jne   short _k1x ; 03/12/2014
2592                                    ;
2593                                    ; 03/12/2014
2594                                    ; 28/08/2014
2595                                    ; PERFORM OTHER FUNCTION ?? here !
2596                                    ;; MOV AX, 9002h           ; MOVE IN WAIT CODE & TYPE
2597                                    ;; INT       15H                 ; PERFORM OTHER FUNCTION
2598                                  _K1T:                              ; ASCII READ
2599 0000109D FB                        sti                     ; INTERRUPTS BACK ON DURING LOOP
2600 0000109E 90                        nop                     ; ALLOW AN INTERRUPT TO OCCUR
2601                                  _K1U:
2602 0000109F FA                        cli                     ; INTERRUPTS BACK OFF
2603 000010A0 8B1D[0C190000]              mov        ebx, [BUFFER_HEAD]  ; GET POINTER TO HEAD OF BUFFER
2604 000010A6 3B1D[10190000]              cmp     ebx, [BUFFER_TAIL]     ; TEST END OF BUFFER
2605                                  _k1x:
2606 000010AC 53                        push  ebx               ; SAVE ADDRESS
2607 000010AD 9C                        pushf                   ; SAVE FLAGS
2608 000010AE E8FD060000                call  MAKE_LED          ; GO GET MODE INDICATOR DATA BYTE
2609 000010B3 8A1D[01190000]            mov   bl, [KB_FLAG_2]   ; GET PREVIOUS BITS
2610 000010B9 30C3                      xor   bl, al            ; SEE IF ANY DIFFERENT
2611 000010BB 80E307                    and   bl, 07h      ; KB_LEDS    ; ISOLATE INDICATOR BITS
2612 000010BE 7406                      jz    short _K1V        ; IF NO CHANGE BYPASS UPDATE
2613 000010C0 E897060000                call  SND_LED1
2614 000010C5 FA                        cli                     ; DISABLE INTERRUPTS
2615                                  _K1V:
2616 000010C6 9D                        popf                    ; RESTORE FLAGS
2617 000010C7 5B                        pop   ebx               ; RESTORE ADDRESS
2618 000010C8 74D3                        je      short _K1T              ; LOOP UNTIL SOMETHING IN BUFFER
2619                                    ;
2620 000010CA 668B03                    mov   ax, [ebx]         ; GET SCAN CODE AND ASCII CODE
2621 000010CD E871000000                  call    _K4                     ; MOVE POINTER TO NEXT POSITION
2622 000010D2 891D[0C190000]              mov     [BUFFER_HEAD], ebx      ; STORE VALUE IN VARIABLE
2623 000010D8 C3                        retn                    ; RETURN
2624
2625                                    ;----- READ THE KEY TO SEE IF ONE IS PRESENT -----
2626                                  _K2S:
2627 000010D9 FA                        cli                     ; INTERRUPTS OFF
```

```
2628 000010DA 8B1D[0C190000]                mov     ebx, [BUFFER_HEAD]      ; GET HEAD POINTER
2629 000010E0 3B1D[10190000]                cmp     ebx, [BUFFER_TAIL]      ; IF EQUAL (Z=1) THEN NOTHING THERE
2630 000010E6 668B03                      mov   ax, [ebx]
2631 000010E9 9C                          pushf                         ; SAVE FLAGS
2632 000010EA 6650                        push  ax                      ; SAVE CODE
2633 000010EC E8BF060000                  call  MAKE_LED                ; GO GET MODE INDICATOR DATA BYTE
2634 000010F1 8A1D[01190000]              mov   bl, [KB_FLAG_2]         ; GET PREVIOUS BITS
2635 000010F7 30C3                        xor   bl, al                  ; SEE IF ANY DIFFERENT
2636 000010F9 80E307                      and   bl, 07h ; KB_LEDS       ; ISOLATE INDICATOR BITS
2637 000010FC 7405                        jz    short _K2T              ; IF NO CHANGE BYPASS UPDATE
2638 000010FE E842060000                  call  SND_LED                       ; GO TURN ON MODE INDICATORS
2639                                  _K2T:
2640 00001103 6658                        pop   ax                      ; RESTORE CODE
2641 00001105 9D                          popf                          ; RESTORE FLAGS
2642 00001106 FB                          sti                           ; INTERRUPTS BACK ON
2643 00001107 C3                          retn                          ; RETURN
2644
2645                                      ;-----  ROUTINE TO TRANSLATE SCAN CODE PAIRS FOR EXTENDED CALLS -----
2646                                  _KIO_E_XLAT:
2647 00001108 3CF0                        cmp   al, 0F0h                ; IS IT ONE OF THE FILL-INs?
2648 0000110A 7506                        jne   short _KIO_E_RET        ; NO, PASS IT ON
2649 0000110C 08E4                          or  ah, ah                  ; AH = 0 IS SPECIAL CASE
2650 0000110E 7402                          jz  short _KIO_E_RET           ; PASS THIS ON UNCHANGED
2651 00001110 30C0                        xor   al, al                  ; OTHERWISE SET AL = 0
2652                                  _KIO_E_RET:
2653 00001112 C3                          retn                          ; GO BACK
2654
2655                                      ;-----  ROUTINE TO TRANSLATE SCAN CODE PAIRS FOR STANDARD CALLS -----
2656                                  _KIO_S_XLAT:
2657 00001113 80FCE0                      cmp   ah, 0E0h                ; IS IT KEYPAD ENTER OR / ?
2658 00001116 750F                        jne   short _KIO_S2           ; NO, CONTINUE
2659 00001118 3C0D                        cmp   al, 0Dh                       ; KEYPAD ENTER CODE?
2660 0000111A 7408                          je  short _KIO_S1           ; YES, MASSAGE A BIT
2661 0000111C 3C0A                        cmp   al, 0Ah                       ; CTRL KEYPAD ENTER CODE?
2662 0000111E 7404                          je  short _KIO_S1           ; YES, MASSAGE THE SAME
2663 00001120 B435                        mov   ah, 35h                       ; NO, MUST BE KEYPAD /
2664                                  _kio_ret: ; 03/12/2014
2665 00001122 F8                          clc
2666 00001123 C3                          retn
2667                                      ;jmp  short _KIO_USE                ; GIVE TO CALLER
2668                                  _KIO_S1:
2669 00001124 B41C                        mov   ah, 1Ch                       ; CONVERT TO COMPATIBLE OUTPUT
2670                                      ;jmp  short _KIO_USE                ; GIVE TO CALLER
2671 00001126 C3                          retn
2672                                  _KIO_S2:
2673 00001127 80FC84                      cmp   ah, 84h                       ; IS IT ONE OF EXTENDED ONES?
2674 0000112A 7715                        ja    short _KIO_DIS                ; YES, THROW AWAY AND GET ANOTHER CHAR
2675 0000112C 3CF0                        cmp   al, 0F0h                ; IS IT ONE OF THE FILL-INs?
2676 0000112E 7506                          jne short _KIO_S3           ; NO, TRY LAST TEST
```

```
2677 00001130 08E4                      or      ah, ah              ; AH = 0 IS SPECIAL CASE
2678 00001132 740C                        jz    short _KIO_USE            ; PASS THIS ON UNCHANGED
2679 00001134 EB0B                      jmp     short _KIO_DIS            ; THROW AWAY THE REST
2680                                  _KIO_S3:
2681 00001136 3CE0                      cmp     al, 0E0h            ; IS IT AN EXTENSION OF A PREVIOUS ONE?
2682                                    ;jne    short _KIO_USE            ; NO, MUST BE A STANDARD CODE
2683 00001138 75E8                      jne     short _kio_ret
2684 0000113A 08E4                      or      ah, ah              ; AH = 0 IS SPECIAL CASE
2685 0000113C 7402                        jz    short _KIO_USE            ; JUMP IF AH = 0
2686 0000113E 30C0                      xor     al, al              ; CONVERT TO COMPATIBLE OUTPUT
2687                                    ;jmp    short _KIO_USE            ; PASS IT ON TO CALLER
2688                                  _KIO_USE:
2689                                    ;clc                        ; CLEAR CARRY TO INDICATE GOOD CODE
2690 00001140 C3                        retn                        ; RETURN
2691                                  _KIO_DIS:
2692 00001141 F9                        stc                         ; SET CARRY TO INDICATE DISCARD CODE
2693 00001142 C3                        retn                        ; RETURN
2694
2695                                    ;----- INCREMENT BUFFER POINTER ROUTINE -----
2696                                  _K4:
2697 00001143 43                        inc     ebx
2698 00001144 43                        inc     ebx                 ; MOVE TO NEXT WORD IN LIST
2699 00001145 3B1D[08190000]              cmp     ebx, [BUFFER_END]     ; AT END OF BUFFER?
2700                                      ;jne    short _K5               ; NO, CONTINUE
2701 0000114B 7206                      jb      short _K5
2702 0000114D 8B1D[04190000]              mov     ebx, [BUFFER_START]     ; YES, RESET TO BUFFER BEGINNING
2703                                  _K5:
2704 00001153 C3                        retn
2705
2706                                  ; 20/02/2015
2707                                  ; 05/12/2014
2708                                  ; 26/08/2014
2709                                  ; KEYBOARD (HARDWARE) INTERRUPT -  IRQ LEVEL 1
2710                                  ; (INT_09h - Retro UNIX 8086 v1 - U9.ASM, 07/03/2014)
2711                                  ;
2712                                  ; Derived from "KB_INT_1" procedure of IBM "pc-at"
2713                                  ; rombios source code (06/10/1985)
2714                                  ; 'keybd.asm', HARDWARE INT 09h - (IRQ Level 1)
2715
2716                                  ;--------- 8042 COMMANDS --------------------------------------------------------
2717                                  ENA_KBD           equ   0AEh   ; ENABLE KEYBOARD COMMAND
2718                                  DIS_KBD           equ   0ADh   ; DISABLE KEYBOARD COMMAND
2719                                  SHUT_CMD    equ   0FEh   ; CAUSE A SHUTDOWN COMMAND
2720                                  ;--------- 8042 KEYBOARD INTERFACE AND DIAGNOSTIC CONTROL REGISTERS ------------
2721                                  STATUS_PORT equ   064h   ; 8042 STATUS PORT
2722                                  INPT_BUF_FULL     equ   00000010b ; 1 = +INPUT BUFFER FULL
2723                                  PORT_A            equ   060h   ; 8042 KEYBOARD SCAN CODE/CONTROL PORT
2724                                  ;---------- 8042 KEYBOARD RESPONSE -------------------------------------------
2725                                  KB_ACK            equ   0FAh   ; ACKNOWLEDGE PROM TRANSMISSION
```

```
KB_RESEND    equ    0FEh   ; RESEND REQUEST
KB_OVER_RUN  equ    0FFh   ; OVER RUN SCAN CODE
;---------- KEYBOARD/LED COMMANDS ----------------------------------------------
KB_ENABLE    equ    0F4h          ; KEYBOARD ENABLE
LED_CMD              equ    0EDh          ; LED WRITE COMMAND
KB_TYPA_RD   equ    0F3h          ; TYPAMATIC RATE/DELAY COMMAND
;---------- KEYBOARD SCAN CODES ------------------------------------------------
NUM_KEY              equ    69            ; SCAN CODE FOR      NUMBER LOCK KEY
SCROLL_KEY   equ    70            ; SCAN CODE FOR      SCROLL LOCK KEY
ALT_KEY              equ    56            ; SCAN CODE FOR      ALTERNATE SHIFT KEY
CTL_KEY              equ    29            ; SCAN CODE FOR      CONTROL KEY
CAPS_KEY     equ    58            ; SCAN CODE FOR      SHIFT LOCK KEY
DEL_KEY              equ    83            ; SCAN CODE FOR      DELETE KEY
INS_KEY              equ    82            ; SCAN CODE FOR      INSERT KEY
LEFT_KEY     equ    42            ; SCAN CODE FOR      LEFT SHIFT
RIGHT_KEY    equ    54            ; SCAN CODE FOR      RIGHT SHIFT
SYS_KEY              equ    84            ; SCAN CODE FOR      SYSTEM KEY
;---------- ENHANCED KEYBOARD SCAN CODES ---------------------------------------
ID_1         equ    0ABh          ; 1ST ID CHARACTER FOR KBX
ID_2         equ    041h          ; 2ND ID CHARACTER FOR KBX
ID_2A        equ    054h          ; ALTERNATE 2ND ID CHARACTER FOR KBX
F11_M        equ    87            ; F11 KEY MAKE
F12_M        equ    88            ; F12 KEY MAKE
MC_E0        equ    224           ; GENERAL MARKER CODE
MC_E1        equ    225           ; PAUSE KEY MARKER CODE
;---------- FLAG EQUATES WITHIN @KB_FLAG----------------------------------------
RIGHT_SHIFT  equ    00000001b     ; RIGHT SHIFT KEY DEPRESSED
LEFT_SHIFT   equ    00000010b     ; LEFT SHIFT KEY DEPRESSED
CTL_SHIFT    equ    00000100b     ; CONTROL SHIFT KEY DEPRESSED
ALT_SHIFT    equ    00001000b     ; ALTERNATE SHIFT KEY DEPRESSED
SCROLL_STATE equ    00010000b     ; SCROLL LOCK STATE IS ACTIVE
NUM_STATE    equ    00100000b     ; NUM LOCK STATE IS ACTIVE
CAPS_STATE   equ    01000000b     ; CAPS LOCK STATE IS ACTIVE
INS_STATE    equ    10000000b     ; INSERT STATE IS ACTIVE
;---------- FLAG EQUATES WITHIN @KB_FLAG_1 -------------------------------------
L_CTL_SHIFT  equ    00000001b     ; LEFT CTL KEY DOWN
L_ALT_SHIFT  equ    00000010b     ; LEFT ALT KEY DOWN
SYS_SHIFT    equ    00000100b     ; SYSTEM KEY DEPRESSED AND HELD
HOLD_STATE   equ    00001000b     ; SUSPEND KEY HAS BEEN TOGGLED
SCROLL_SHIFT equ    00010000b     ; SCROLL LOCK KEY IS DEPRESSED
NUM_SHIFT    equ    00100000b     ; NUM LOCK KEY IS DEPRESSED
CAPS_SHIFT   equ    01000000b     ; CAPS LOCK KEY IS DEPRE55ED
INS_SHIFT    equ    10000000b     ; INSERT KEY IS DEPRESSED
;---------- FLAGS EQUATES WITHIN @KB_FLAG_2 ------------------------------------
KB_LEDS              equ    00000111b     ; KEYBOARD LED STATE BITS
;            equ    00000001b     ; SCROLL LOCK INDICATOR
;            equ    00000010b     ; NUM LOCK INDICATOR
;            equ    00000100b     ; CAPS LOCK INDICATOR
;            equ    00001000b     ; RESERVED (MUST BE ZERO)
```

```
                                KB_FA       equ    00010000b    ; ACKNOWLEDGMENT RECEIVED
                                KB_FE       equ    00100000b    ; RESEND RECEIVED FLAG
                                KB_PR_LED   equ    01000000b    ; MODE INDICATOR UPDATE
                                KB_ERR              equ    10000000b    ; KEYBOARD TRANSMIT ERROR FLAG
                                ;---------- FLAGS EQUATES WITHIN @KB_FLAG_3 ----------------------------------
                                LC_E1       equ    00000001b    ; LAST CODE WAS THE E1 HIDDEN CODE
                                LC_E0       equ    00000010b    ; LAST CODE WAS THE E0 HIDDEN CODE
                                R_CTL_SHIFT equ    00000100b    ; RIGHT CTL KEY DOWN
                                R_ALT_SHIFT equ    00001000b    ; RIGHT ALT KEY DOWN
                                GRAPH_ON    equ    00001000b    ; ALT GRAPHICS KEY DOWN (WT ONLY)
                                KBX         equ    00010000b    ; ENHANCED KEYBOARD INSTALLED
                                SET_NUM_LK  equ    00100000b    ; FORCE NUM LOCK IF READ ID AND KBX
                                LC_AB       equ    01000000b    ; LAST CHARACTER WAS FIRST ID CHARACTER
                                RD_ID       equ    10000000b    ; DOING A READ ID (MUST BE BIT0)
                                ;
                                ;----------- INTERRUPT EQUATES -----------------------------------------------
                                EOI         equ    020h         ; END OF INTERRUPT COMMAND TO 8259
                                INTA00              equ    020h         ; 8259 PORT


                                kb_int:

                                ; 05/12/2014
                                ; 04/12/2014 (derivation from pc-xt-286 bios source code -1986-,
                                      ;           instead of pc-at bios - 1985-)
                                ; 26/08/2014
                                ;
                                ; 03/06/86  KEYBOARD BIOS
                                ;
                                ;--- HARDWARE INT 09H -- (IRQ LEVEL 1) ---------------------------------------
                                ;                                                                           ;
                                ;     KEYBOARD INTERRUPT ROUTINE                                            ;
                                ;                                                                           ;
                                ;---------------------------------------------------------------------------

                                KB_INT_1:
00001154 FB                           sti                       ; ENABLE INTERRUPTS
                                      ;push  ebp
00001155 50                           push   eax
00001156 53                           push   ebx
00001157 51                           push   ecx
00001158 52                           push   edx
00001159 56                           push   esi
0000115A 57                           push   edi
0000115B 1E                           push   ds
0000115C 06                           push   es
0000115D FC                           cld                       ; FORWARD DIRECTION
0000115E 66B81000                     mov    ax, KDATA
00001162 8ED8                         mov    ds, ax
```

```
2824 00001164 8EC0                      mov   es, ax
2825                                    ;
2826                                    ;----- WAIT FOR KEYBOARD DISABLE COMMAND TO BE ACCEPTED
2827 00001166 B0AD                      mov   al, DIS_KBD         ; DISABLE THE KEYBOARD COMMAND
2828 00001168 E877050000                call  SHIP_IT                   ; EXECUTE DISABLE
2829 0000116D FA                        cli                       ; DISABLE INTERRUPTS
2830 0000116E B900000100                mov   ecx, 10000h         ; SET MAXIMUM TIMEOUT
2831                                KB_INT_01:
2832 00001173 E464                      in    al, STATUS_PORT           ; READ ADAPTER STATUS
2833 00001175 A802                      test  al, INPT_BUF_FULL   ; CHECK INPUT BUFFER FULL STATUS BIT
2834 00001177 E0FA                      loopnz KB_INT_01          ; WAIT FOR COMMAND TO BE ACCEPTED
2835                                    ;
2836                                    ;----- READ CHARACTER FROM KEYBOARD INTERFACE
2837 00001179 E460                      in    al, PORT_A          ; READ IN THE CHARACTER
2838                                    ;
2839                                    ;----- SYSTEM HOOK INT 15H - FUNCTION 4FH (ON HARDWARE INT LEVEL 9H)
2840                                    ;MOV  AH, 04FH            ; SYSTEM INTERCEPT - KEY CODE FUNCTION
2841                                    ;STC                      ; SET CY=1 (IN CASE OF IRET)
2842                                    ;INT  15H                 ; CASETTE CALL (AL)=KEY SCAN CODE
2843                                    ;                         ; RETURNS CY=1 FOR INVALID FUNCTION
2844                                    ;JC   KB_INT_02           ; CONTINUE IF CARRY FLAG SET ((AL)=CODE)
2845                                    ;JMP  K26                 ; EXIT IF SYSTEM HANDLES SCAN CODE
2846                                    ;                         ; EX¦T HANDLES HARDWARE EOI AND ENABLE
2847                                    ;
2848                                    ;----- CHECK FOR A RESEND COMMAND TO KEYBOARD
2849                                KB_INT_02:                    ;       (AL)= SCAN CODE
2850 0000117B FB                        sti                       ; ENABLE INTERRUPTS AGAIN
2851 0000117C 3CFE                      cmp   al, KB_RESEND       ; IS THE INPUT A RESEND
2852 0000117E 7411                        je    short KB_INT_4          ; GO IF RESEND
2853                                    ;
2854                                    ;----- CHECK FOR RESPONSE TO A COMMAND TO KEYBOARD
2855 00001180 3CFA                      cmp   al, KB_ACK          ; IS THE INPUT AN ACKNOWLEDGE
2856 00001182 751A                        jne   short KB_INT_2          ; GO IF NOT
2857                                    ;
2858                                    ;----- A COMMAND TO THE KEYBOARD WAS ISSUED
2859 00001184 FA                        cli                       ; DISABLE INTERRUPTS
2860 00001185 800D[01190000]10          or    byte [KB_FLAG_2], KB_FA ; INDICATE ACK RECEIVED
2861 0000118C E97A020000                  jmp   K26                     ; RETURN IF NOT (ACK RETURNED FOR DATA)
2862                                    ;
2863                                    ;----- RESEND THE LAST BYTE
2864                                KB_INT_4:
2865 00001191 FA                        cli                       ; DISABLE INTERRUPTS
2866 00001192 800D[01190000]20          or    byte [KB_FLAG_2], KB_FE ; INDICATE RESEND RECEIVED
2867 00001199 E96D020000                  jmp   K26                     ; RETURN IF NOT ACK RETURNED FOR DATA)
2868                                    ;
2869                                ;-----      UPDATE MODE INDICATORS IF CHANGE IN STATE
2870                                KB_INT_2:
2871 0000119E 6650                      push  ax                  ; SAVE DATA IN
2872 000011A0 E80B060000                call  MAKE_LED            ; GO GET MODE INDICATOR DATA BYTE
```

```
2873 000011A5 8A1D[01190000]              mov   bl, [KB_FLAG_2]     ; GET PREVIOUS BITS
2874 000011AB 30C3                        xor   bl, al              ; SEE IF ANY DIFFERENT
2875 000011AD 80E307                      and   bl, KB_LEDS         ; ISOLATE INDICATOR BITS
2876 000011B0 7405                        jz    short UP0           ; IF NO CHANGE BYPASS UPDATE
2877 000011B2 E88E050000                  call  SND_LED                   ; GO TURN ON MODE INDICATORS
2878                                UP0:
2879 000011B7 6658                        pop   ax                  ; RESTORE DATA IN
2880                                ;--------------------------------------------------------------------
2881                                ;     START OF KEY PROCESSING                                        ;
2882                                ;--------------------------------------------------------------------
2883 000011B9 88C4                        mov   ah, al              ; SAVE SCAN CODE IN AH ALSO
2884                                      ;
2885                                      ;----- TEST FOR OVERRUN SCAN CODE FROM KEYBOARD
2886 000011BB 3CFF                        cmp   al, KB_OVER_RUN           ; IS THIS AN OVERRUN CHAR
2887 000011BD 0F840D050000                  je      K62                   ; BUFFER_FULL_BEEP
2888                                      ;
2889                                K16:
2890 000011C3 8A3D[02190000]              mov   bh, [KB_FLAG_3]           ; LOAD FLAGS FOR TESTING
2891                                      ;
2892                                      ;----- TEST TO SEE IF A READ_ID IS IN PROGRESS
2893 000011C9 F6C7C0                      test  bh, RD_ID+LC_AB     ; ARE WE DOING A READ ID?
2894 000011CC 7449                        jz    short NOT_ID        ; CONTINUE IF NOT
2895 000011CE 7917                        jns   short TST_ID_2            ; IS THE RD_ID FLAG ON?
2896 000011D0 3CAB                        cmp   al, ID_1            ; IS THIS THE 1ST ID CHARACTER?
2897 000011D2 7507                        jne   short RST_RD_ID
2898 000011D4 800D[02190000]40            or    byte [KB_FLAG_3], LC_AB ; INDICATE 1ST ID WAS OK
2899                                RST_RD_ID:
2900 000011DB 8025[02190000]7F            and   byte [KB_FLAG_3], ~RD_ID ; RESET THE READ ID FLAG
2901                                        ;jmp    short ID_EX           ; AND EXIT
2902 000011E2 E924020000                  jmp   K26
2903                                      ;
2904                                TST_ID_2:
2905 000011E7 8025[02190000]BF            and   byte [KB_FLAG_3], ~LC_AB ; RESET FLAG
2906 000011EE 3C54                        cmp   al, ID_2A           ; IS THIS THE 2ND ID CHARACTER?
2907 000011F0 7419                          je  short KX_BIT        ; JUMP IF SO
2908 000011F2 3C41                        cmp   al, ID_2            ; IS THIS THE 2ND ID CHARACTER?
2909                                        ;jne short ID_EX        ; LEAVE IF NOT
2910 000011F4 0F8511020000                jne   K26
2911                                      ;
2912                                      ;----- A READ ID SAID THAT IT WAS ENHANCED KEYBOARD
2913 000011FA F6C720                      test  bh, SET_NUM_LK            ; SHOULD WE SET NUM LOCK?
2914 000011FD 740C                          jz      short KX_BIT          ; EXIT IF NOT
2915 000011FF 800D[FF180000]20            or    byte [KB_FLAG], NUM_STATE ; FORCE NUM LOCK ON
2916 00001206 E83A050000                  call  SND_LED                   ; GO SET THE NUM LOCK INDICATOR
2917                                KX_BIT:
2918 0000120B 800D[02190000]10            or    byte [KB_FLAG_3], KBX     ; INDICATE ENHANCED KEYBOARD WAS FOUND
2919 00001212 E9F4010000            ID_EX:      jmp     K26               ; EXIT
2920                                      ;
2921                                NOT_ID:
```

```
2922 00001217 3CE0                      cmp    al, MC_E0           ; IS THIS THE GENERAL MARKER CODE?
2923 00001219 750C                      jne    short TEST_E1
2924 0000121B 800D[02190000]12          or     byte [KB_FLAG_3], LC_E0+KBX ; SET FLAG BIT, SET KBX, AND
2925                                    ;jmp   short EXIT          ; THROW AWAY THIS CODE
2926 00001222 E9EB010000                jmp    K26A
2927                                TEST_E1:
2928 00001227 3CE1                      cmp    al, MC_E1           ; IS THIS THE PAUSE KEY?
2929 00001229 750C                      jne    short NOT_HC
2930 0000122B 800D[02190000]11          or     byte [KB_FLAG_3], LC_E1+KBX ; SET FLAG BIT, SET KBX, AND
2931 00001232 E9DB010000            EXIT: jmp  K26A                ; THROW AWAY THIS CODE
2932                                    ;
2933                                NOT_HC:
2934 00001237 247F                      and    al, 07Fh            ; TURN OFF THE BREAK BIT
2935 00001239 F6C702                    test   bh, LC_E0           ; LAST CODE THE E0 MARKER CODE
2936 0000123C 7414                      jz     short NOT_LC_E0           ; JUMP IF NOT
2937                                    ;
2938 0000123E BF[E7170000]              mov    edi, _K6+6          ; IS THIS A SHIFT KEY?
2939 00001243 AE                        scasb
2940                                    ;je    short K16B          ; YES, THROW AWAY & RESET FLAG
2941 00001244 0F84C1010000              je     K26
2942 0000124A AE                        scasb
2943 0000124B 757C                      jne    short K16A          ; NO, CONTINUE KEY PROCESSING
2944                                    ;jmp   short K16B
2945 0000124D E9B9010000                jmp    K26
2946                                    ;
2947                                NOT_LC_E0:
2948 00001252 F6C701                    test   bh, LC_E1           ; LAST CODE THE E1 MARKER CODE?
2949 00001255 7435                      jz     short T_SYS_KEY           ; JUMP IF NOT
2950 00001257 B904000000                mov    ecx, 4              ; LENGHT OF SEARCH
2951 0000125C BF[E5170000]              mov    edi, _K6+4          ; IS THIS AN ALT, CTL, OR SHIFT?
2952 00001261 F2AE                      repne  scasb               ; CHECK IT
2953                                    ;je    short EXIT          ; THROW AWAY IF SO
2954 00001263 0F84A9010000              je     K26A
2955                                    ;
2956 00001269 3C45                      cmp    al, NUM_KEY         ; IS IT THE PAUSE KEY?
2957                                    ;jne   short K16B          ; NO, THROW AWAY & RESET FLAG
2958 0000126B 0F859A010000              jne    K26
2959 00001271 F6C480                    test   ah, 80h                   ; YES, IS IT THE BREAK OF THE KEY?
2960                                    ;jnz   short K16B          ; YES, THROW THIS AWAY, TOO
2961 00001274 0F8591010000              jnz    K26
2962                                      ; 20/02/2015
2963 0000127A F605[00190000]08          test   byte [KB_FLAG_1],HOLD_STATE ;  NO, ARE WE PAUSED ALREADY?
2964                                    ;jnz   short K16B          ;  YES, THROW AWAY
2965 00001281 0F8584010000              jnz    K26
2966 00001287 E9D3020000                jmp     K39P                    ; NO, THIS IS THE REAL PAUSE STATE
2967                                    ;
2968                                    ;----- TEST FOR SYSTEM KEY
2969                                T_SYS_KEY:
2970 0000128C 3C54                      cmp    al, SYS_KEY         ; IS IT THE SYSTEM KEY?
```

```
2971 0000128E 7539                      jnz    short K16A          ; CONTINUE IF NOT
2972                                    ;
2973 00001290 F6C480                    test   ah, 80h                   ; CHECK IF THIS A BREAK CODE
2974 00001293 7524                      jnz    short K16C          ; DO NOT TOUCH SYSTEM INDICATOR IF TRUE
2975                                    ;
2976 00001295 F605[00190000]04          test   byte [KB_FLAG_1], SYS_SHIFT ; SEE IF IN SYSTEM KEY HELD DOWN
2977                                      ;jnz short K16B          ; IF YES, DO NOT PROCESS SYSTEM INDICATOR
2978 0000129C 0F8569010000              jnz     K26
2979                                    ;
2980 000012A2 800D[00190000]04          or     byte [KB_FLAG_1], SYS_SHIFT ; INDICATE SYSTEM KEY DEPRESSED
2981 000012A9 B020                      mov    al, EOI                   ; END OF INTERRUPT COMMAND
2982 000012AB E620                      out    20h, al ;out INTA00, al   ; SEND COMMAND TO INTERRUPT CONTROL PORT
2983                                                                     ; INTERRUPT-RETURN-NO-EOI
2984 000012AD B0AE                      mov    al, ENA_KBD         ; INSURE KEYBOARD IS ENABLED
2985 000012AF E830040000                call   SHIP_IT                   ; EXECUTE ENABLE
2986                                    ; !!! SYSREQ !!! function/system call (INTERRUPT) must be here !!!
2987                                    ;MOV   AL, 8500H           ; FUNCTION VALUE FOR MAKE OF SYSTEM KEY
2988                                    ;STI                       ; MAKE SURE INTERRUPTS ENABLED
2989                                    ;INT   15H                 ; USER INTERRUPT
2990 000012B4 E965010000                  jmp     K27A                    ; END PROCESSING
2991                                    ;
2992                                ;K16B:      jmp    K26                ; IGNORE SYSTEM KEY
2993                                    ;
2994                                K16C:
2995 000012B9 8025[00190000]FB              and    byte [KB_FLAG_1], ~SYS_SHIFT ; TURN OFF SHIFT KEY HELD DOWN
2996 000012C0 B020                          mov    al, EOI                   ; END OF INTERRUPT COMMAND
2997 000012C2 E620                          out    20h, al ;out INTA00, al ; SEND COMMAND TO INTERRUPT CONTROL PORT
2998                                                                         ; INTERRUPT-RETURN-NO-EOI
2999                                        ;MOV   AL, ENA_KBD         ; INSURE KEYBOARD IS ENABLED
3000                                        ;CALL  SHIP_IT                   ; EXECUTE ENABLE
3001                                        ;
3002                                        ;MOV   AX, 8501H           ; FUNCTION VALUE FOR BREAK OF SYSTEM KEY
3003                                        ;STI                       ; MAKE SURE INTERRUPTS ENABLED
3004                                        ;INT   15H                 ; USER INTERRUPT
3005                                        ;JMP   K27A                ; INGONRE SYSTEM KEY
3006                                        ;
3007 000012C4 E94E010000                    jmp     K27                ; IGNORE SYSTEM KEY
3008                                        ;
3009                                        ;----- TEST FOR SHIFT KEYS
3010                                K16A:
3011 000012C9 8A1D[FF180000]                mov    bl, [KB_FLAG]       ; PUT STATE FLAGS IN BL
3012 000012CF BF[E1170000]                  mov    edi, _K6            ; SHIFT KEY TABLE offset
3013 000012D4 B908000000                    mov    ecx, _K6L           ; LENGTH
3014 000012D9 F2AE                          repne  scasb               ; LOOK THROUGH THE TABLE FOR A MATCH
3015 000012DB 88E0                          mov    al, ah              ; RECOVER SCAN CODE
3016 000012DD 0F8510010000                    jne     K25                    ; IF NO MATCH, THEN SHIFT NOT FOUND
3017                                        ;
3018                                        ;------      SHIFT KEY FOUND
3019                                K17:
```

```
3020 000012E3 81EF[E2170000]                sub     edi, _K6+1              ; ADJUST PTR TO SCAN CODE MATCH
3021 000012E9 8AA7[E9170000]                     mov     ah, [edi+_K7]           ; GET MASK INTO AH
3022 000012EF B102                          mov    cl, 2                ; SETUP COUNT FOR FLAG SHIFTS
3023 000012F1 A880                          test   al, 80h                    ; TEST FOR BREAK KEY
3024 000012F3 0F8596000000                    jnz     K23                     ; JUMP OF BREAK
3025                                        ;
3026                                        ;----- SHIFT MAKE FOUND, DETERMINE SET OR TOGGLE
3027                                   K17C:
3028 000012F9 80FC10                        cmp    ah, SCROLL_SHIFT
3029 000012FC 732B                          jae    short K18           ; IF SCROLL SHIFT OR ABOVE, TOGGLE KEY
3030                                        ;
3031                                        ;----- PLAIN SHIFT KEY, SET SHIFT ON
3032 000012FE 0825[FF180000]                or     [KB_FLAG], ah       ; TURN ON SHIFT BIT
3033 00001304 A80C                            test al, CTL_SHIFT+ALT_SHIFT ; IS IT ALT OR CTRL?
3034                                        ;jnz   short K17D          ; YES, MORE FLAGS TO SET
3035 00001306 0F84FF000000                  jz     K26                 ; NO, INTERRUPT RETURN
3036                                   K17D:
3037 0000130C F6C702                        test   bh, LC_E0           ; IS THIS ONE OF NEW KEYS?
3038 0000130F 740B                          jz     short K17E          ; NO, JUMP
3039 00001311 0825[02190000]                or     [KB_FLAG_3], ah           ; SET BITS FOR RIGHT CTRL, ALT
3040 00001317 E9EF000000                    jmp    K26                 ; INTERRUPT RETURN
3041                                   K17E:
3042 0000131C D2EC                          shr    ah, cl              ; MOVE FLAG BITS TWO POSITIONS
3043 0000131E 0825[00190000]                or     [KB_FLAG_1], ah           ; SET BITS FOR LEFT CTRL, ALT
3044 00001324 E9E2000000                    jmp    K26
3045                                        ;
3046                                        ;----- TOGGLED SHIFT KEY, TEST FOR 1ST MAKE OR NOT
3047                                   K18:                            ; SHIFT-TOGGLE
3048 00001329 F6C304                        test   bl, CTL_SHIFT              ; CHECK CTL SHIFT STATE
3049                                          ;jz        short K18A              ; JUMP IF NOT CTL STATE
3050 0000132C 0F85C1000000                    jnz     K25                     ; JUMP IF CTL STATE
3051                                   K18A:
3052 00001332 3C52                          cmp    al, INS_KEY         ; CHECK FOR INSERT KEY
3053 00001334 7524                          jne    short K22           ; JUMP IF NOT INSERT KEY
3054 00001336 F6C308                        test   bl, ALT_SHIFT              ; CHECK FOR ALTERNATE SHIFT
3055                                               ;jz    short K18B          ; JUMP IF NOT ALTERNATE SHIFT
3056 00001339 0F85B4000000                    jnz     K25                     ; JUMP IF ALTERNATE SHIFT
3057                                   K18B:
3058 0000133F F6C702                        test   bh, LC_E0 ;20/02/2015      ; IS THIS NEW INSERT KEY?
3059 00001342 7516                          jnz    short K22           ; YES, THIS ONE'S NEVER A '0'
3060                                   K19:
3061 00001344 F6C320                        test   bl, NUM_STATE              ; CHECK FOR BASE STATE
3062 00001347 750C                          jnz    short K21           ; JUMP IF NUM LOCK IS ON
3063 00001349 F6C303                        test   bl, LEFT_SHIFT+RIGHT_SHIFT ; TEST FOR SHIFT STATE
3064 0000134C 740C                          jz     short K22           ; JUMP IF BASE STATE
3065                                   K20:                            ; NUMERIC ZERO, NOT INSERT KEY
3066 0000134E 88C4                          mov    ah, al              ; PUT SCAN CODE BACK IN AH
3067 00001350 E99E000000                      jmp     K25                     ; NUMERAL '0', STNDRD. PROCESSING
3068                                   K21:                            ; MIGHT BE NUMERIC
```

```
3069 00001355 F6C303                    test   bl, LEFT_SHIFT+RIGHT_SHIFT
3070 00001358 74F4                      jz     short K20           ; IS NUMERIC, STD. PROC.
3071                                    ;
3072                                K22:                            ; SHIFT TOGGLE KEY HIT; PROCESS IT
3073 0000135A 8425[00190000]            test   ah, [KB_FLAG_1]     ; IS KEY ALREADY DEPRESSED
3074 00001360 0F85A5000000                jnz     K26                     ; JUMP IF KEY ALREADY DEPRESSED
3075                                K22A:
3076 00001366 0825[00190000]              or      [KB_FLAG_1], ah ; INDICATE THAT THE KEY IS DEPRESSED
3077 0000136C 3025[FF180000]            xor    [KB_FLAG], ah       ; TOGGLE THE SHIFT STATE
3078                                    ;
3079                                    ;----- TOGGLE LED IF CAPS, NUM  OR SCROLL KEY DEPRESSED
3080 00001372 F6C470                    test   ah, CAPS_SHIFT+NUM_SHIFT+SCROLL_SHIFT ; SHIFT TOGGLE?
3081 00001375 7409                      jz     short K22B          ; GO IF NOT
3082                                    ;
3083 00001377 6650                      push   ax                  ; SAVE SCAN CODE AND SHIFT MASK
3084 00001379 E8C7030000                call   SND_LED                  ; GO TURN MODE INDICATORS ON
3085 0000137E 6658                      pop    ax                  ; RESTORE SCAN CODE
3086                                K22B:
3087 00001380 3C52                      cmp    al, INS_KEY         ; TEST FOR 1ST MAKE OF INSERT KEY
3088 00001382 0F8583000000                jne     K26                     ; JUMP IF NOT INSERT KEY
3089 00001388 88C4                      mov    ah, al               ; SCAN CODE IN BOTH HALVES OF AX
3090 0000138A E999000000                  jmp     K28                     ; FLAGS UPDATED, PROC. FOR BUFFER
3091                                    ;
3092                                    ;----- BREAK SHIFT FOUND
3093                                K23:                            ; BREAK-SHIFT-FOUND
3094 0000138F 80FC10                    cmp    ah, SCROLL_SHIFT    ; IS THIS A TOGGLE KEY
3095 00001392 F6D4                      not    ah                  ; INVERT MASK
3096 00001394 7355                      jae    short K24           ; YES, HANDLE BREAK TOGGLE
3097 00001396 2025[FF180000]            and    [KB_FLAG], ah       ; TURN OFF SHIFT BIT
3098 0000139C 80FCFB                    cmp    ah, ~CTL_SHIFT           ; IS THIS ALT OR CTL?
3099 0000139F 7730                      ja     short K23D          ; NO, ALL DONE
3100                                    ;
3101 000013A1 F6C702                    test   bh, LC_E0           ; 2ND ALT OR CTL?
3102 000013A4 7408                      jz     short K23A          ; NO, HANSLE NORMALLY
3103 000013A6 2025[02190000]            and    [KB_FLAG_3], ah          ; RESET BIT FOR RIGHT ALT OR CTL
3104 000013AC EB08                      jmp    short K23B          ; CONTINUE
3105                                K23A:
3106 000013AE D2FC                      sar    ah, cl              ; MOVE THE MASK BIT TWO POSITIONS
3107 000013B0 2025[00190000]            and    [KB_FLAG_1], ah          ; RESET BIT FOR LEFT ALT AND CTL
3108                                K23B:
3109 000013B6 88C4                      mov    ah, al              ; SAVE SCAN CODE
3110 000013B8 A0[02190000]              mov    al, [KB_FLAG_3]          ; GET RIGHT ALT & CTRL FLAGS
3111 000013BD D2E8                      shr    al, cl              ; MOVE TO BITS 1 & 0
3112 000013BF 0A05[00190000]            or     al, [KB_FLAG_1]          ; PUT IN LEFT ALÌT & CTL FLAGS
3113 000013C5 D2E0                      shl    al, cl              ; MOVE BACK TO BITS 3 & 2
3114 000013C7 240C                      and    al, ALT_SHIFT+CTL_SHIFT ; FILTER OUT OTHER GARBAGE
3115 000013C9 0805[FF180000]            or     [KB_FLAG], al       ; PUT RESULT IN THE REAL FLAGS
3116 000013CF 88E0                      mov    al, ah
3117                                K23D:
```

```
 3118 000013D1 3CB8                     cmp   al, ALT_KEY+80h           ; IS THIS ALTERNATE SHIFT RELEASE
 3119 000013D3 7536                     jne   short K26          ; INTERRUPT RETURN
 3120                                   ;
 3121                                   ;----- ALTERNATE SHIFT KEY RELEASED, GET THE VALUE INTO BUFFER
 3122 000013D5 A0[03190000]             mov   al, [ALT_INPUT]
 3123 000013DA B400                     mov   ah, 0              ; SCAN CODE OF 0
 3124 000013DC 8825[03190000]           mov   [ALT_INPUT], ah    ; ZERO OUT THE FIELD
 3125 000013E2 3C00                     cmp   al, 0              ; WAS THE INPUT = 0?
 3126 000013E4 7425                     je    short K26          ; INTERRUPT_RETURN
 3127 000013E6 E9C2020000                 jmp    K61                    ; IT WASN'T, SO PUT IN BUFFER
 3128                                   ;
 3129                                K24:                              ; BREAK-TOGGLE
 3130 000013EB 2025[00190000]           and   [KB_FLAG_1], ah    ; INDICATE NO LONGER DEPRESSED
 3131 000013F1 EB18                     jmp   short K26          ; INTERRUPT_RETURN
 3132                                   ;
 3133                                   ;----- TEST FOR HOLD STATE
 3134                                                            ; AL, AH = SCAN CODE
 3135                                K25:                              ; NO-SHIFT-FOUND
 3136 000013F3 3C80                     cmp   al, 80h                  ; TEST FOR BREAK KEY
 3137 000013F5 7314                     jae   short K26          ; NOTHING FOR BREAK CHARS FROM HERE ON
 3138 000013F7 F605[00190000]08         test  byte [KB_FLAG_1], HOLD_STATE ; ARE WE IN HOLD STATE
 3139 000013FE 7428                     jz    short K28          ; BRANCH AROUND TEST IF NOT
 3140 00001400 3C45                     cmp   al, NUM_KEY
 3141 00001402 7407                     je    short K26          ; CAN'T END HOLD ON NUM_LOCK
 3142 00001404 8025[00190000]F7         and   byte [KB_FLAG_1], ~HOLD_STATE ; TURN OFF THE HOLD STATE BIT
 3143                                   ;
 3144                                K26:
 3145 0000140B 8025[02190000]FC         and   byte [KB_FLAG_3], ~(LC_E0+LC_E1) ; RESET LAST CHAR H.C. FLAG
 3146                                K26A:                             ; INTERRUPT-RETURN
 3147 00001412 FA                       cli                       ; TURN OFF INTERRUPTS
 3148 00001413 B020                     mov   al, EOI                  ; END OF INTERRUPT COMMAND
 3149 00001415 E620                     out   20h, al      ;out INTA00, al     ; SEND COMMAND TO INTERRUPT CONTROL
PORT
 3150                                K27:                              ; INTERRUPT-RETURN-NO-EOI
 3151 00001417 B0AE                     mov   al, ENA_KBD         ; INSURE KEYBOARD IS ENABLED
 3152 00001419 E8C6020000               call  SHIP_IT                  ; EXECUTE ENABLE
 3153                                K27A:
 3154 0000141E FA                       cli                       ; DISABLE INTERRUPTS
 3155 0000141F 07                       pop   es                  ; RESTORE REGISTERS
 3156 00001420 1F                       pop   ds
 3157 00001421 5F                       pop   edi
 3158 00001422 5E                       pop   esi
 3159 00001423 5A                       pop   edx
 3160 00001424 59                       pop   ecx
 3161 00001425 5B                       pop   ebx
 3162 00001426 58                       pop   eax
 3163                                   ;pop  ebp
 3164 00001427 CF                       iret                      ; RETURN
 3165
```

```
3166                                    ;----- NOT IN HOLD STATE
3167                            K28:                            ; NO-HOLD-STATE
3168 00001428 3C58                      cmp   al, 88            ; TEST FOR OUT-OF-RANGE SCAN CODES
3169 0000142A 77DF                      ja    short K26         ; IGNORE IF OUT-OF-RANGE
3170                                    ;
3171 0000142C F6C308                    test  bl, ALT_SHIFT           ; ARE WE IN ALTERNATE SHIFT
3172                                      ;jz short K28A        ; IF NOT ALTERNATE
3173 0000142F 0F84E8000000                jz    K38
3174                                    ;
3175 00001435 F6C710                    test  bh, KBX                 ; IS THIS THE ENCHANCED KEYBOARD?
3176                                    ;jz   short K29         ; NO, ALT STATE IS REAL
3177 00001438 0F85DF000000                jnz   K38                     ; YES, THIS IS PHONY ALT STATE
3178                                      ;                     ; DUE TO PRESSING SYSREQ
3179                            ;K28A:      jmp   short K38
3180                                    ;
3181                                    ;----- TEST FOR RESET KEY SEQUENCE (CTL ALT DEL)
3182                            K29:                            ; TEST-RESET
3183 0000143E F6C304                    test  bl, CTL_SHIFT           ; ARE WE IN CONTROL SHIFT ALSO?
3184 00001441 740B                      jz    short K31         ; NO_RESET
3185 00001443 3C53                      cmp   al, DEL_KEY       ; CTL-ALT STATE, TEST FOR DELETE KEY
3186 00001445 7507                      jne   short K31         ; NO_RESET, IGNORE
3187                                    ;
3188                                    ;----- CTL-ALT-DEL HAS BEEN FOUND
3189                                    ; 26/08/2014
3190                            cpu_reset:
3191                                    ; IBM PC/AT ROM BIOS source code - 10/06/85 (TEST4.ASM - PROC_SHUTDOWN)
3192                                    ; Send FEh (system reset command) to the keyboard controller.
3193 00001447 B0FE                      mov   al, SHUT_CMD      ; SHUTDOWN COMMAND
3194 00001449 E664                      out   STATUS_PORT, al         ; SEND TO KEYBOARD CONTROL PORT
3195                            khere:
3196 0000144B F4                        hlt                     ; WAIT FOR 80286 RESET
3197 0000144C EBFD                      jmp   short khere       ; INSURE HALT
3198
3199                                    ;
3200                                    ;----- IN ALTERNATE SHIFT, RESET NOT FOUND
3201                            K31:                            ; NO-RESET
3202 0000144E 3C39                      cmp   al, 57            ; TEST FOR SPACE KEY
3203 00001450 7507                      jne   short K311        ; NOT THERE
3204 00001452 B020                      mov   al, ' '                 ; SET SPACE CHAR
3205 00001454 E943020000                  jmp   K57                   ; BUFFER_FILL
3206                            K311:
3207 00001459 3C0F                      cmp   al, 15            ; TEST FOR TAB KEY
3208 0000145B 7509                      jne   short K312        ; NOT THERE
3209 0000145D 66B800A5                  mov   ax, 0A500h        ; SET SPECIAL CODE FOR ALT-TAB
3210 00001461 E936020000                  jmp   K57                   ; BUFFER_FILL
3211                            K312:
3212 00001466 3C4A                      cmp   al, 74            ; TEST FOR KEY PAD -
3213 00001468 0F84A2000000                je    K37B                  ; GO PROCESS
3214 0000146E 3C4E                      cmp   al, 78            ; TEST FOR KEY PAD +
```

```
3215 00001470 0F849A000000                      je      K37B                    ; GO PROCESS
3216                                            ;
3217                                            ;----- LOOK FOR KEY PAD ENTRY
3218                                    K32:                                    ; ALT-KEY-PAD
3219 00001476 BF[BD170000]                      mov     edi, K30                ; ALT-INPUT-TABLE offset
3220 0000147B B90A000000                        mov     ecx, 10                         ; LOOK FOR ENTRY USING KEYPAD
3221 00001480 F2AE                              repne   scasb                   ; LOOK FOR MATCH
3222 00001482 7525                              jne     short K33              ; NO_ALT_KEYPAD
3223 00001484 F6C702                            test    bh, LC_E0               ; IS THIS ONE OF THE NEW KEYS?
3224 00001487 0F858A000000                      jnz     K37C                            ; YES, JUMP, NOT NUMPAD KEY
3225 0000148D 81EF[BE170000]                    sub     edi, K30+1              ; DI NOW HAS ENTRY VALUE
3226 00001493 A0[03190000]                      mov     al, [ALT_INPUT]         ; GET THE CURRENT BYTE
3227 00001498 B40A                              mov     ah, 10                  ; MULTIPLY BY 10
3228 0000149A F6E4                              mul     ah
3229 0000149C 6601F8                            add     ax, di                  ; ADD IN THE LATEST ENTRY
3230 0000149F A2[03190000]                      mov     [ALT_INPUT], al         ; STORE IT AWAY
3231                                    ;K32A:
3232 000014A4 E962FFFFFF                        jmp     K26                             ; THROW AWAY THAT KEYSTROKE
3233                                            ;
3234                                            ;----- LOOK FOR SUPERSHIFT ENTRY
3235                                    K33:                                    ; NO-ALT-KEYPAD
3236 000014A9 C605[03190000]00                  mov     byte [ALT_INPUT], 0     ; ZERO ANY PREVIOUS ENTRY INTO INPUT
3237 000014B0 B91A000000                        mov     ecx, 26                         ; (DI),(ES) ALREADY POINTING
3238 000014B5 F2AE                              repne   scasb                   ; LOOK FOR MATCH IN ALPHABET
3239 000014B7 7450                              je      short K37A              ; MATCH FOUND, GO FILLL THE BUFFER
3240                                            ;
3241                                            ;----- LOOK FOR TOP ROW OF ALTERNATE SHIFT
3242                                    K34:                                    ; ALT-TOP-ROW
3243 000014B9 3C02                              cmp     al, 2                   ; KEY WITH '1' ON IT
3244 000014BB 7253                              jb      short K37B              ; MUST BE ESCAPE
3245 000014BD 3C0D                              cmp     al, 13                  ; IS IT IN THE REGION
3246 000014BF 7705                              ja      short K35               ; NO, ALT SOMETHING ELSE
3247 000014C1 80C476                            add     ah, 118                         ; CONVERT PSEUDO SCAN CODE TO RANGE
3248 000014C4 EB43                              jmp     short K37A              ; GO FILL THE BUFFER
3249                                            ;
3250                                            ;----- TRANSLATE ALTERNATE SHIFT PSEUDO SCAN CODES
3251                                    K35:                                    ; ALT-FUNCTION
3252 000014C6 3C57                              cmp     al, F11_M               ; IS IT F11?
3253 000014C8 7209                              jb      short K35A ; 20/02/2015    ; NO, BRANCH
3254 000014CA 3C58                              cmp     al, F12_M               ; IS IT F12?
3255 000014CC 7705                              ja      short K35A ; 20/02/2015    ; NO, BRANCH
3256 000014CE 80C434                            add     ah, 52                  ; CONVERT TO PSEUDO SCAN CODE
3257 000014D1 EB36                              jmp     short K37A              ; GO FILL THE BUFFER
3258                                    K35A:
3259 000014D3 F6C702                            test    bh, LC_E0               ; DO WE HAVE ONE OF THE NEW KEYS?
3260 000014D6 7422                              jz      short K37               ; NO, JUMP
3261 000014D8 3C1C                              cmp     al, 28                  ; TEST FOR KEYPAD ENTER
3262 000014DA 7509                              jne     short K35B                      ; NOT THERE
3263 000014DC 66B800A6                          mov     ax, 0A600h              ; SPECIAL CODE
```

```
3264 000014E0 E9B7010000                jmp    K57                ; BUFFER FILL
3265                                K35B:
3266 000014E5 3C53                      cmp    al, 83             ; TEST FOR DELETE KEY
3267 000014E7 742E                      je     short K37C         ; HANDLE WITH OTHER EDIT KEYS
3268 000014E9 3C35                      cmp    al, 53             ; TEST FOR KEYPAD /
3269                                    ;jne   short K32A         ; NOT THERE, NO OTHER E0 SPECIALS
3270 000014EB 0F851AFFFFFF                jne     K26
3271 000014F1 66B800A4                  mov    ax, 0A400h         ; SPECIAL CODE
3272 000014F5 E9A2010000                jmp    K57                ; BUFFER FILL
3273                                K37:
3274 000014FA 3C3B                      cmp    al, 59             ; TEST FOR FUNCTION KEYS (F1)
3275 000014FC 7212                        jb     short K37B            ; NO FN, HANDLE W/OTHER EXTENDED
3276 000014FE 3C44                      cmp    al, 68             ; IN KEYPAD REGION?
3277                                      ;ja  short K32A         ; IF SO, IGNORE
3278 00001500 0F8705FFFFFF                ja     K26
3279 00001506 80C42D                    add    ah, 45             ; CONVERT TO PSEUDO SCAN CODE
3280                                K37A:
3281 00001509 B000                      mov    al, 0              ; ASCII CODE OF ZERO
3282 0000150B E98C010000                  jmp     K57                    ; PUT IT IN THE BUFFER
3283                                K37B:
3284 00001510 B0F0                      mov    al, 0F0h           ; USE SPECIAL ASCII CODE
3285 00001512 E985010000                jmp     K57                   ; PUT IT IN THE BUFFER
3286                                K37C:
3287 00001517 0450                      add    al, 80             ; CONVERT SCAN CODE (EDIT KEYS)
3288 00001519 88C4                      mov    ah, al             ; (SCAN CODE NOT IN AH FOR INSERT)
3289 0000151B EBEC                      jmp     short K37A             ; PUT IT IN THE BUFFER
3290                                    ;
3291                                    ;----- NOT IN ALTERNATE SHIFT
3292                                K38:                              ; NOT-ALT-SHIFT
3293                                                                  ; BL STILL HAS SHIFT FLAGS
3294 0000151D F6C304                    test   bl, CTL_SHIFT             ; ARE WE IN CONTROL SHIFT?
3295                                    ;jnz   short K38A         ; YES, START PROCESSING
3296 00001520 0F84AB000000                jz     K44                    ; NOT-CTL-SHIFT
3297                                    ;
3298                                    ;----- CONTROL SHIFT, TEST SPECIAL CHARACTERS
3299                                    ;----- TEST FOR BREAK
3300                                K38A:
3301 00001526 3C46                      cmp    al, SCROLL_KEY             ; TEST FOR BREAK
3302 00001528 752C                      jne    short K39          ; JUMP, NO-BREAK
3303 0000152A F6C710                    test   bh, KBX                   ; IS THIS THE ENHANCED KEYBOARD?
3304 0000152D 7405                      jz     short K38B         ; NO, BREAK IS VALID
3305 0000152F F6C702                    test   bh, LC_E0          ; YES, WAS LAST CODE AN E0?
3306 00001532 7422                      jz     short K39          ; NO-BREAK, TEST FOR PAUSE
3307                                K38B:
3308 00001534 8B1D[0C190000]            mov    ebx, [BUFFER_HEAD]  ; RESET BUFFER TO EMPTY
3309 0000153A 891D[10190000]            mov    [BUFFER_TAIL], ebx
3310 00001540 C605[FE180000]80          mov    byte [BIOS_BREAK], 80h  ; TURN ON BIOS_BREAK BIT
3311                                    ;
3312                                    ;----- ENABLE KEYBOARD
```

```
3313 00001547 B0AE                      mov   al, ENA_KBD         ; ENABLE KEYBOARD
3314 00001549 E896010000                call  SHIP_IT                   ; EXECUTE ENABLE
3315                                    ;
3316                                    ; CTRL+BREAK code here !!!
3317                                    ;INT  1BH                 ; BREAK INTERRUPT VECTOR
3318                                    ;
3319 0000154E 6629C0                    sub   ax, ax              ; PUT OUT DUMMY CHARACTER
3320 00001551 E946010000                  jmp     K57                     ; BUFFER_FILL
3321                                    ;
3322                                    ;----- TEST FOR PAUSE
3323                                K39:                          ; NO_BREAK
3324 00001556 F6C710                    test  bh, KBX                   ; IS THIS THE ENHANCED KEYBOARD?
3325 00001559 7537                      jnz   short K41           ; YES, THEN THIS CAN'T BE PAUSE
3326 0000155B 3C45                      cmp   al, NUM_KEY         ; LOOK FOR PAUSE KEY
3327 0000155D 7533                      jne   short K41           ; NO-PAUSE
3328                                K39P:
3329 0000155F 800D[00190000]08          or    byte [KB_FLAG_1], HOLD_STATE ; TURN ON THE HOLD FLAG
3330                                    ;
3331                                    ;----- ENABLE KEYBOARD
3332 00001566 B0AE                      mov   al, ENA_KBD         ; ENABLE KEYBOARD
3333 00001568 E877010000                call  SHIP_IT                   ; EXECUTE ENABLE
3334                                K39A:
3335 0000156D B020                      mov   al, EOI                   ; END OF INTERRUPT TO CONTROL PORT
3336 0000156F E620                      out   20h, al ;out INTA00, al   ; ALLOW FURTHER KEYSTROKE INTERRUPTS
3337                                    ;
3338                                    ;----- DURING PAUSE INTERVAL, TURN COLOR CRT BACK ON
3339 00001571 803D[FC180000]07            cmp     byte [CRT_MODE], 7      ; IS THIS BLACK AND WHITE CARD
3340 00001578 740A                        je      short K40                       ; YES, NOTHING TO DO
3341 0000157A 66BAD803                  mov   dx, 03D8h           ; PORT FOR COLOR CARD
3342 0000157E A0[FD180000]                mov     al, [CRT_MODE_SET]     ; GET THE VALUE OF THE CURRENT MODE
3343 00001583 EE                        out   dx, al              ; SET THE CRT MODE, SO THAT CRT IS ON
3344                                    ;
3345                                K40:                          ; PAUSE-LOOP
3346 00001584 F605[00190000]08            test    byte [KB_FLAG_1], HOLD_STATE ; CHECK HOLD STATE FLAG
3347 0000158B 75F7                      jnz   short K40           ; LOOP UNTIL FLAG TURNED OFF
3348                                    ;
3349 0000158D E985FEFFFF                  jmp     K27                     ; INTERRUPT_RETURN_NO_EOI
3350                                      ;
3351                                    ;----- TEST SPECIAL CASE KEY 55
3352                                K41:                          ; NO-PAUSE
3353 00001592 3C37                      cmp   al, 55              ; TEST FOR */PRTSC KEY
3354 00001594 7513                      jne   short K42           ; NOT-KEY-55
3355 00001596 F6C710                    test  bh, KBX                   ; IS THIS THE ENHANCED KEYBOARD?
3356 00001599 7405                      jz    short K41A          ; NO, CTL-PRTSC IS VALID
3357 0000159B F6C702                    test  bh, LC_E0           ; YES, WAS LAST CODE AN E0?
3358 0000159E 7421                      jz    short K42B          ; NO, TRANSLATE TO A FUNCTION
3359                                K41A:
3360 000015A0 66B80072                  mov   ax, 114*256         ; START/STOP PRINTING SWITCH
3361 000015A4 E9F3000000                  jmp     K57                     ; BUFFER_FILL
```

```
3362                                    ;
3363                                    ;----- SET UP TO TRANSLATE CONTROL SHIFT
3364                            K42:                            ; NOT-KEY-55
3365 000015A9 3C0F                     cmp     al, 15          ; IS IT THE TAB KEY?
3366 000015AB 7414                     je      short K42B      ; YES, XLATE TO FUNCTION CODE
3367 000015AD 3C35                     cmp     al, 53          ; IS IT THE / KEY?
3368 000015AF 750E                     jne     short K42A      ; NO, NO MORE SPECIAL CASES
3369 000015B1 F6C702                   test    bh, LC_E0       ; YES, IS IT FROM THE KEY PAD?
3370 000015B4 7409                     jz      short K42A      ; NO, JUST TRANSLATE
3371 000015B6 66B80095                 mov     ax, 9500h       ; YES, SPECIAL CODE FOR THIS ONE
3372 000015BA E9DD000000               jmp     K57             ; BUFFER FILL
3373                            K42A:
3374                                    ;;mov  ebx, _K8        ; SET UP TO TRANSLATE CTL
3375 000015BF 3C3B                     cmp     al, 59          ; IS IT IN CHARACTER TABLE?
3376                                      ;jb   short K45F          ; YES, GO TRANSLATE CHAR
3377                                    ;;jb    K56 ; 20/02/2015
3378                                    ;;jmp   K64 ; 20/02/2015
3379                            K42B:
3380 000015C1 BB[F1170000]             mov     ebx, _K8        ; SET UP TO TRANSLATE CTL
3381 000015C6 0F82AE000000             jb      K56 ;; 20/02/2015
3382 000015CC E9B9000000               jmp     K64
3383                                      ;
3384                                    ;----- NOT IN CONTROL SHIFT
3385                            K44:                            ; NOT-CTL-SHIFT
3386 000015D1 3C37                     cmp     al, 55          ; PRINT SCREEN KEY?
3387 000015D3 7528                     jne     short K45       ; NOT PRINT SCREEN
3388 000015D5 F6C710                   test    bh, KBX                 ; IS THIS ENHANCED KEYBOARD?
3389 000015D8 7407                     jz      short K44A      ; NO, TEST FOR SHIFT STATE
3390 000015DA F6C702                   test    bh, LC_E0       ; YES, LAST CODE A MARKER?
3391 000015DD 7507                     jnz     short K44B      ; YES, IS PRINT SCREEN
3392 000015DF EB41                     jmp     short K45C      ; NO, TRANSLATE TO '*' CHARACTER
3393                            K44A:
3394 000015E1 F6C303                   test    bl, LEFT_SHIFT+RIGHT_SHIFT ; NOT 101 KBD, SHIFT KEY DOWN?
3395 000015E4 743C                     jz      short K45C      ; NO, TRANSLATE TO '*' CHARACTER
3396                                    ;
3397                                    ;----- ISSUE INTERRUPT TO INDICATE PRINT SCREEN FUNCTION
3398                            K44B:
3399 000015E6 B0AE                     mov     al, ENA_KBD     ; INSURE KEYBOARD IS ENABLED
3400 000015E8 E8F7000000               call    SHIP_IT                 ; EXECUTE ENABLE
3401 000015ED B020                     mov     al, EOI                 ; END OF CURRENT INTERRUPT
3402 000015EF E620                     out     20h, al ;out INTA00, al ; SO FURTHER THINGS CAN HAPPEN
3403                                    ; Print Screen !!!      ; ISSUE PRINT SCREEN INTERRUPT (INT 05h)
3404                                    ;PUSH  BP               ; SAVE POINTER
3405                                    ;INT   5H               ; ISSUE PRINT SCREEN INTERRUPT
3406                                    ;POP   BP               ; RESTORE POINTER
3407 000015F1 8025[02190000]FC           and    byte [KB_FLAG_3], ~(LC_E0+LC_E1) ; ZERO OUT THESE FLAGS
3408 000015F8 E91AFEFFFF                 jmp    K27                      ; GO BACK WITHOUT EOI OCCURRING
3409                                    ;
3410                                    ;----- HANDLE IN-CORE KEYS
```

```
3411                                    K45:                            ; NOT-PRINT-SCREEN
3412 000015FD 3C3A                              cmp     al, 58          ; TEST FOR IN-CORE AREA
3413 000015FF 7734                              ja      short K46       ; JUMP IF NOT
3414 00001601 3C35                              cmp     al, 53          ; IS THIS THE '/' KEY?
3415 00001603 7505                              jne     short K45A      ; NO, JUMP
3416 00001605 F6C702                            test    bh, LC_E0       ; WAS THE LAST CODE THE MARKER?
3417 00001608 7518                              jnz     short K45C      ; YES, TRANSLATE TO CHARACTER
3418                                    K45A:
3419 0000160A B91A000000                        mov     ecx, 26                 ; LENGHT OF SEARCH
3420 0000160F BF[C7170000]                      mov     edi, K30+10     ; POINT TO TABLE OF A-Z CHARS
3421 00001614 F2AE                              repne   scasb           ; IS THIS A LETTER KEY?
3422                                            ; 20/02/2015
3423 00001616 7505                              jne     short K45B          ; NO, SYMBOL KEY
3424                                            ;
3425 00001618 F6C340                            test    bl, CAPS_STATE          ; ARE WE IN CAPS_LOCK?
3426 0000161B 750C                              jnz     short K45D      ; TEST FOR SURE
3427                                    K45B:
3428 0000161D F6C303                            test    bl, LEFT_SHIFT+RIGHT_SHIFT ; ARE WE IN SHIFT STATE?
3429 00001620 750C                              jnz     short K45E      ; YES, UPPERCASE
3430                                                                    ; NO, LOWERCASE
3431                                    K45C:
3432 00001622 BB[49180000]                      mov     ebx, K10        ; TRANSLATE TO LOWERCASE LETTERS
3433 00001627 EB51                              jmp     short K56
3434                                    K45D:                           ; ALMOST-CAPS-STATE
3435 00001629 F6C303                            test    bl, LEFT_SHIFT+RIGHT_SHIFT ; CL ON. IS SHIFT ON, TOO?
3436 0000162C 75F4                              jnz     short K45C      ; SHIFTED TEMP OUT OF CAPS STATE
3437                                    K45E:
3438 0000162E BB[A1180000]                      mov     ebx, K11        ; TRANSLATE TO UPPER CASE LETTERS
3439 00001633 EB45                      K45F:   jmp     short K56
3440                                            ;
3441                                            ;----- TEST FOR KEYS F1 - F10
3442                                    K46:                            ; NOT IN-CORE AREA
3443 00001635 3C44                              cmp     al, 68          ; TEST FOR F1 - F10
3444                                            ;ja     short K47       ; JUMP IF NOT
3445                                            ;jmp    short K53       ; YES, GO DO FN KEY PROCESS
3446 00001637 7635                              jna     short K53
3447                                            ;
3448                                            ;----- HANDLE THE NUMERIC PAD KEYS
3449                                    K47:                            ; NOT F1 - F10
3450 00001639 3C53                              cmp     al, 83          ; TEST NUMPAD KEYS
3451 0000163B 772D                              ja      short K52       ; JUMP IF NOT
3452                                            ;
3453                                            ;----- KEYPAD KEYS, MUST TEST NUM LOCK FOR DETERMINATION
3454                                    K48:
3455 0000163D 3C4A                              cmp     al , 74                 ; SPECIAL CASE FOR MINUS
3456 0000163F 74ED                              je      short K45E      ; GO TRANSLATE
3457 00001641 3C4E                              cmp     al , 78                 ; SPECIAL CASE FOR PLUS
3458 00001643 74E9                              je      short K45E      ; GO TRANSLATE
3459 00001645 F6C702                            test    bh, LC_E0       ; IS THIS ONE OFTHE NEW KEYS?
```

```
3460 00001648 750A                      jnz    short K49          ; YES, TRANSLATE TO BASE STATE
3461                                    ;
3462 0000164A F6C320                    test   bl, NUM_STATE      ; ARE WE IN NUM LOCK
3463 0000164D 7514                      jnz    short K50          ; TEST FOR SURE
3464 0000164F F6C303                    test   bl, LEFT_SHIFT+RIGHT_SHIFT ; ARE WE IN SHIFT STATE?
3465                                    ;jnz   short K51          ; IF SHIFTED, REALLY NUM STATE
3466 00001652 75DA                      jnz    short K45E
3467                                    ;
3468                                    ;----- BASE CASE FOR KEYPAD
3469                                K49:
3470 00001654 3C4C                      cmp    al, 76             ; SPECIAL CASE FOR BASE STATE 5
3471 00001656 7504                      jne    short K49A         ; CONTINUE IF NOT KEYPAD 5
3472 00001658 B0F0                      mov    al, 0F0h           ; SPECIAL ASCII CODE
3473 0000165A EB40                      jmp    short K57          ; BUFFER FILL
3474                                K49A:
3475 0000165C BB[49180000]              mov    ebx, K10           ; BASE CASE TABLE
3476 00001661 EB27                      jmp    short K64          ; CONVERT TO PSEUDO SCAN
3477                                    ;
3478                                    ;----- MIGHT BE NUM LOCK, TEST SHIFT STATUS
3479                                K50:                           ; ALMOST-NUM-STATE
3480 00001663 F6C303                      test    bl, LEFT_SHIFT+RIGHT_SHIFT
3481 00001666 75EC                      jnz    short K49          ; SHIFTED TEMP OUT OF NUM STATE
3482 00001668 EBC4                  K51: jmp   short K45E         ; REALLY NUM STATE
3483                                    ;
3484                                    ;----- TEST FOR THE NEW KEYS ON WT KEYBOARDS
3485                                K52:                           ; NOT A NUMPAD KEY
3486 0000166A 3C56                      cmp    al, 86             ; IS IT THE NEW WT KEY?
3487                                    ;jne   short K53          ; JUMP IF NOT
3488                                    ;jmp   short K45B         ; HANDLE WITH REST OF LETTER KEYS
3489 0000166C 74AF                      je     short K45B
3490                                    ;
3491                                    ;----- MUST BE F11 OR F12
3492                                K53:                           ; F1 - F10 COME HERE, TOO
3493 0000166E F6C303                    test   bl, LEFT_SHIFT+RIGHT_SHIFT ; TEST SHIFT STATE
3494 00001671 74E1                      jz     short K49          ; JUMP, LOWER CASE PSEUDO SC'S
3495                                           ; 20/02/2015
3496 00001673 BB[A1180000]              mov    ebx, K11           ; UPPER CASE PSEUDO SCAN CODES
3497 00001678 EB10                      jmp    short K64          ; TRANSLATE SCAN
3498                                    ;
3499                                    ;----- TRANSLATE THE CHARACTER
3500                                K56:                           ; TRANSLATE-CHAR
3501 0000167A FEC8                      dec    al                 ; CONVERT ORIGIN
3502 0000167C D7                        xlat                      ; CONVERT THE SCAN CODE TO ASCII
3503 0000167D F605[02190000]02          test   byte [KB_FLAG_3], LC_E0   ; IS THIS A NEW KEY?
3504 00001684 7416                      jz     short K57          ; NO, GO FILL BUFFER
3505 00001686 B4E0                      mov    ah, MC_E0          ; YES, PUT SPECIAL MARKER IN AH
3506 00001688 EB12                      jmp    short K57          ; PUT IT INTO THE BUFFER
3507                                    ;
3508                                    ;----- TRANSLATE SCAN FOR PSEUDO SCAN CODES
```

```
3509                                    K64:                            ; TRANSLATE-SCAN-ORGD
3510 0000168A FEC8                              dec     al              ; CONVERT ORIGIN
3511 0000168C D7                                xlat                            ; CTL TABLE SCAN
3512 0000168D 88C4                              mov     ah, al          ; PUT VALUE INTO AH
3513 0000168F B000                              mov     al, 0           ; ZERO ASCII CODE
3514 00001691 F605[02190000]02                  test    byte [KB_FLAG_3], LC_E0   ; IS THIS A NEW KEY?
3515 00001698 7402                              jz      short K57       ; NO, GO FILL BUFFER
3516 0000169A B0E0                              mov     al, MC_E0       ; YES, PUT SPECIAL MARKER IN AL
3517                                            ;
3518                                            ;----- PUT CHARACTER INTO BUFFER
3519                                    K57:                            ; BUFFER_FILL
3520 0000169C 3CFF                              cmp     al, -1          ; IS THIS AN IGNORE CHAR
3521                                              ;je   short K59       ; YES, DO NOTHING WITH IT
3522 0000169E 0F8467FDFFFF                      je        K26           ; YES, DO NOTHING WITH IT
3523 000016A4 80FCFF                            cmp     ah, -1          ; LOOK FOR -1 PSEUDO SCAN
3524                                              ;jne  short K61       ; NEAR_INTEERUPT_RETURN
3525 000016A7 0F845EFDFFFF                      je        K26           ; INTERRUPT_RETURN
3526                                    ;K59:                           ; NEAR_INTEERUPT_RETURN
3527                                    ;       jmp     K26             ; INTERRUPT_RETURN
3528                                    K61:                            ; NOT-CAPS-STATE
3529 000016AD 8B1D[10190000]                    mov     ebx, [BUFFER_TAIL] ; GET THE END POINTER TO THE BUFFER
3530 000016B3 89DE                              mov     esi, ebx        ; SAVE THE VALUE
3531 000016B5 E889FAFFFF                        call    _K4             ; ADVANCE THE TAIL
3532 000016BA 3B1D[0C190000]                    cmp     ebx, [BUFFER_HEAD] ; HAS THE BUFFER WRAPPED AROUND
3533 000016C0 740E                              je      short K62       ; BUFFER_FULL_BEEP
3534 000016C2 668906                            mov     [esi], ax       ; STORE THE VALUE
3535 000016C5 891D[10190000]                    mov     [BUFFER_TAIL], ebx ; MOVE THE POINTER UP
3536 000016CB E93BFDFFFF                        jmp     K26
3537                                            ;;cli                   ; TURN OFF INTERRUPTS
3538                                            ;;mov   al, EOI                 ; END OF INTERRUPT COMMAND
3539                                            ;;out   INTA00, al      ; SEND COMMAND TO INTERRUPT CONTROL PORT
3540                                            ;MOV    AL, ENA_KBD     ; INSURE KEYBOARD IS ENABLED
3541                                            ;CALL   SHIP_IT                 ; EXECUTE ENABLE
3542                                            ;MOV    AX, 9102H       ; MOVE IN POST CODE & TYPE
3543                                            ;INT    15H             ; PERFORM OTHER FUNCTION
3544                                            ;;and   byte [KB_FLAG_3],~(LC_E0+LC_E1) ; RESET LAST CHAR H.C. FLAG
3545                                            ;JMP    K27A            ; INTERRUPT_RETURN
3546                                            ;;jmp    K27
3547                                            ;
3548                                            ;----- BUFFER IS FULL SOUND THE BEEPER
3549                                    K62:
3550 000016D0 B020                              mov     al, EOI                 ; ENABLE INTERRUPT CONTROLLER CHIP
3551 000016D2 E620                              out     INTA00, al
3552 000016D4 66B9A602                          mov     cx, 678                 ; DIVISOR FOR 1760 HZ
3553 000016D8 B304                              mov     bl, 4           ; SHORT BEEP COUNT (1/16 + 1/64 DELAY)
3554 000016DA E82A030000                        call    beep            ; GO TO COMMON BEEP HANDLER
3555 000016DF E933FDFFFF                        jmp      K27            ; EXIT
3556
3557                                    SHIP_IT:
```

```
                                        ;--------------------------------------------------------------------------
----
                                        ; SHIP_IT
                                        ;       THIS ROUTINES HANDLES TRANSMISSION OF COMMAND AND DATA BYTES
                                        ;       TO THE KEYBOARD CONTROLLER.
                                        ;--------------------------------------------------------------------------
----
                                        ;
000016E4 6650                           push    ax                      ; SAVE DATA TO SEND
                                        ;
                                        ;-----  WAIT FOR COMMAND TO ACCEPTED
000016E6 FA                             cli                             ; DISABLE INTERRUPTS TILL DATA SENT
                                        ; xor   ecx, ecx                ; CLEAR TIMEOUT COUNTER
000016E7 B900000100                     mov     ecx, 10000h
                                S10:
000016EC E464                           in      al, STATUS_PORT                 ; READ KEYBOARD CONTROLLER STATUS
000016EE A802                           test    al, INPT_BUF_FULL       ; CHECK FOR ITS INPUT BUFFER BUSY
000016F0 E0FA                           loopnz  S10                     ; WAIT FOR COMMAND TO BE ACCEPTED
                                        ;
000016F2 6658                           pop     ax                      ; GET DATA TO SEND
000016F4 E664                           out     STATUS_PORT, al                 ; SEND TO KEYBOARD CONTROLLER
000016F6 FB                             sti                             ; ENABLE INTERRUPTS AGAIN
000016F7 C3                             retn                            ; RETURN TO CALLER
                                        ;
                                SND_DATA:
                                        ; --------------------------------------------------------------------------
-----
                                        ; SND_DATA
                                        ;       THIS ROUTINES HANDLES TRANSMISSION OF COMMAND AND DATA BYTES
                                        ;       TO THE KEYBOARD AND RECEIPT OF ACKNOWLEDGEMENTS. IT ALSO
                                        ;       HANDLES ANY RETRIES IF REQUIRED
                                        ; --------------------------------------------------------------------------
-----
                                        ;
000016F8 6650                           push    ax                      ; SAVE REGISTERS
000016FA 6653                           push    bx
000016FC 51                             push    ecx
000016FD 88C7                           mov     bh, al                  ; SAVE TRANSMITTED BYTE FOR RETRIES
000016FF B303                           mov     bl, 3                   ; LOAD RETRY COUNT
                                SD0:
00001701 FA                             cli                             ; DISABLE INTERRUPTS
00001702 8025[01190000]CF               and     byte [KB_FLAG_2], ~(KB_FE+KB_FA) ; CLEAR ACK AND RESEND FLAGS
                                        ;
                                        ;-----  WAIT FOR COMMAND TO BE ACCEPTED
00001709 B900000100                     mov     ecx, 10000h             ; MAXIMUM WAIT COUNT
                                SD5:
0000170E E464                           in      al, STATUS_PORT                 ; READ KEYBOARD PROCESSOR STATUS PORT
00001710 A802                           test    al, INPT_BUF_FULL       ; CHECK FOR ANY PENDING COMMAND
00001712 E0FA                           loopnz  SD5                     ; WAIT FOR COMMAND TO BE ACCEPTED
```

```
                                        ;
00001714 88F8                           mov     al, bh              ; REESTABLISH BYTE TO TRANSMIT
00001716 E660                           out     PORT_A, al          ; SEND BYTE
00001718 FB                             sti                         ; ENABLE INTERRUPTS
                                        ;mov    cx, 01A00h          ; LOAD COUNT FOR 10 ms+
00001719 B9FFFF0000                     mov     ecx, 0FFFFh
                                SD1:
0000171E F605[01190000]30               test    byte [KB_FLAG_2], KB_FE+KB_FA ; SEE IF EITHER BIT SET
00001725 750F                           jnz     short SD3           ; IF SET, SOMETHING RECEIVED GO PROCESS
00001727 E2F5                           loop    SD1                 ; OTHERWISE WAIT
                                SD2:
00001729 FECB                           dec     bl                  ; DECREMENT RETRY COUNT
0000172B 75D4                           jnz     short SD0           ; RETRY TRANSMISSION
0000172D 800D[01190000]80               or      byte [KB_FLAG_2], KB_ERR ; TURN ON TRANSMIT ERROR FLAG
00001734 EB09                           jmp     short SD4           ; RETRIES EXHAUSTED FORGET TRANSMISSION
                                SD3:
00001736 F605[01190000]10               test    byte [KB_FLAG_2], KB_FA ; SEE IF THIS IS AN ACKNOWLEDGE
0000173D 74EA                           jz      short SD2           ; IF NOT, GO RESEND
                                SD4:
0000173F 59                             pop     ecx                 ; RESTORE REGISTERS
00001740 665B                           pop     bx
00001742 6658                           pop     ax
00001744 C3                             retn                        ; RETURN, GOOD TRANSMISSION

                                SND_LED:
                                        ; --------------------------------------------------------------------------
-----
                                        ; SND_LED
                                        ;       THIS ROUTINES TURNS ON THE MODE INDICATORS.
                                        ;
                                        ;--------------------------------------------------------------------------
-----
                                        ;
00001745 FA                             cli                         ; TURN OFF INTERRUPTS
00001746 F605[01190000]40               test    byte [KB_FLAG_2], KB_PR_LED ; CHECK FOR MODE INDICATOR UPDATE
0000174D 755F                           jnz     short SL1           ; DON'T UPDATE AGAIN IF UPDATE UNDERWAY
                                        ;
0000174F 800D[01190000]40               or      byte [KB_FLAG_2], KB_PR_LED ; TURN ON UPDATE IN PROCESS
00001756 B020                           mov     al, EOI                     ; END OF INTERRUPT COMMAND
00001758 E620                           out     20h, al ;out INTA00, al     ; SEND COMMAND TO INTERRUPT CONTROL PORT
0000175A EB11                           jmp     short SL0           ; GO SEND MODE INDICATOR COMMAND
                                SND_LED1:
0000175C FA                             cli                         ; TURN OFF INTERRUPTS
0000175D F605[01190000]40               test    byte [KB_FLAG_2], KB_PR_LED ; CHECK FOR MODE INDICATOR UPDATE
00001764 7548                           jnz     short SL1           ; DON'T UPDATE AGAIN IF UPDATE UNDERWAY
                                        ;
00001766 800D[01190000]40               or      byte [KB_FLAG_2], KB_PR_LED ; TURN ON UPDATE IN PROCESS
                                SL0:
0000176D B0ED                           mov     al, LED_CMD         ; LED CMD BYTE
```

```
  3650 0000176F E884FFFFFF                call  SND_DATA           ; SEND DATA TO KEYBOARD
  3651 00001774 FA                        cli
  3652 00001775 E836000000                call  MAKE_LED           ; GO FORM INDICATOR DATA BYTE
  3653 0000177A 8025[01190000]F8          and   byte [KB_FLAG_2], 0F8h    ; ~KB_LEDS ; CLEAR MODE INDICATOR BITS
  3654 00001781 0805[01190000]            or    byte [KB_FLAG_2], al      ; SAVE PRESENT INDICATORS FOR NEXT TIME
  3655 00001787 F605[01190000]80          test  byte [KB_FLAG_2], KB_ERR ; TRANSMIT ERROR DETECTED
  3656 0000178E 750F                      jnz   short SL2          ; IF SO, BYPASS SECOND BYTE TRANSMISSION
  3657                                    ;
  3658 00001790 E863FFFFFF                call  SND_DATA           ; SEND DATA TO KEYBOARD
  3659 00001795 FA                        cli                      ; TURN OFF INTERRUPTS
  3660 00001796 F605[01190000]80          test  byte [KB_FLAG_2], KB_ERR ; TRANSMIT ERROR DETECTED
  3661 0000179D 7408                      jz    short SL3          ; IF NOT, DON'T SEND AN ENABLE COMMAND
  3662                                SL2:
  3663 0000179F B0F4                      mov   al, KB_ENABLE      ; GET KEYBOARD CSA ENABLE COMMAND
  3664 000017A1 E852FFFFFF                call  SND_DATA           ; SEND DATA TO KEYBOARD
  3665 000017A6 FA                        cli                      ; TURN OFF INTERRUPTS
  3666                                SL3:
  3667 000017A7 8025[01190000]3F          and   byte [KB_FLAG_2], ~(KB_PR_LED+KB_ERR) ; TURN OFF MODE INDICATOR
  3668                                SL1:                         ; UPDATE AND TRANSMIT ERROR FLAG
  3669 000017AE FB                        sti                      ; ENABLE INTERRUPTS
  3670 000017AF C3                        retn                     ; RETURN TO CALLER
  3671
  3672                                MAKE_LED:
  3673                                    ;--------------------------------------------------------------------------
----
  3674                                    ; MAKE_LED
  3675                                    ;     THIS ROUTINES FORMS THE DATA BYTE NECESSARY TO TURN ON/OFF
  3676                                    ;     THE MODE INDICATORS.
  3677                                    ;--------------------------------------------------------------------------
----
  3678                                    ;
  3679                                    ;push  cx                ; SAVE CX
  3680 000017B0 A0[FF180000]              mov   al, byte [KB_FLAG]  ; GET CAPS & NUM LOCK INDICATORS
  3681 000017B5 2470                      and   al, CAPS_STATE+NUM_STATE+SCROLL_STATE ; ISOLATE INDICATORS
  3682                                    ;mov  cl, 4              ; SHIFT COUNT
  3683                                    ;rol  al, cl             ; SHIFT BITS OVER TO TURN ON INDICATORS
  3684 000017B7 C0C004                    rol   al, 4 ; 20/02/2015
  3685 000017BA 2407                      and   al, 07h                 ; MAKE SURE ONLY MODE BITS ON
  3686                                    ;pop  cx
  3687 000017BC C3                        retn                     ; RETURN TO CALLER
  3688
  3689                                ;--------------------------------------------------------------------------
  3690                                ;     KEY IDENTIFICATION SCAN TABLES
  3691                                ;--------------------------------------------------------------------------
  3692
  3693                                ;-----      TABLES FOR ALT CASE ------------
  3694                                ;-----      ALT-INPUT-TABLE
  3695 000017BD 524F50514B            K30:  db    82,79,80,81,75
  3696 000017C2 4C4D474849                  db    76,77,71,72,73           ; 10 NUMBER ON KEYPAD
```

```
3697                                    ;-----      SUPER-SHIFT-TABLE
3698 000017C7 101112131415                    db    16,17,18,19,20,21   ; A-Z TYPEWRITER CHARS
3699 000017CD 161718191E1F                    db    22,23,24,25,30,31
3700 000017D3 202122232425                    db    32,33,34,35,36,37
3701 000017D9 262C2D2E2F30                    db    38,44,45,46,47,48
3702 000017DF 3132                            db    49,50
3703
3704                                    ;-----      TABLE OF SHIFT KEYS AND MASK VALUES
3705                                    ;-----      KEY_TABLE
3706 000017E1 52                        _K6:    db      INS_KEY                 ; INSERT KEY
3707 000017E2 3A4546381D                      db    CAPS_KEY,NUM_KEY,SCROLL_KEY,ALT_KEY,CTL_KEY
3708 000017E7 2A36                              db    LEFT_KEY,RIGHT_KEY
3709                                    _K6L    equ   $-_K6
3710
3711                                    ;-----      MASK_TABLE
3712 000017E9 80                        _K7:    db      INS_SHIFT               ; INSERT MODE SHIFT
3713 000017EA 4020100804                      db    CAPS_SHIFT,NUM_SHIFT,SCROLL_SHIFT,ALT_SHIFT,CTL_SHIFT
3714 000017EF 0201                            db    LEFT_SHIFT,RIGHT_SHIFT
3715
3716                                    ;-----      TABLES FOR CTRL CASE            ;---- CHARACTERS ------
3717 000017F1 1BFF00FFFFFF              _K8:  db    27,-1,0,-1,-1,-1    ; Esc, 1, 2, 3, 4, 5
3718 000017F7 1EFFFFFFFF1F                    db    30,-1,-1,-1,-1,31   ; 6, 7, 8, 9, 0, -
3719 000017FD FF7FFF111705                    db    -1,127,-1,17,23,5   ; =, Bksp, Tab, Q, W, E
3720 00001803 12141915090F                    db    18,20,25,21,9,15    ; R, T, Y, U, I, O
3721 00001809 101B1D0AFF01                    db    16,27,29,10,-1,1    ; P, [, ], Enter, Ctrl, A
3722 0000180F 13040607080A                    db    19,4,6,7,8,10       ; S, D, F, G, H, J
3723 00001815 0B0CFFFFFFFF                    db    11,12,-1,-1,-1,-1   ; K, L, :, ', `, LShift
3724 0000181B 1C1A18031602                    db    28,26,24,3,22,2           ; Bkslash, Z, X, C, V, B
3725 00001821 0E0DFFFFFFFF                    db    14,13,-1,-1,-1,-1   ; N, M, ,, ., /, RShift
3726 00001827 96FF20FF                        db    150,-1,' ',-1       ; *, ALT, Spc, CL
3727                                          ;                         ;----- FUNCTIONS ------
3728 0000182B 5E5F60616263                    db    94,95,96,97,98,99   ; F1 - F6
3729 00001831 64656667FFFF                    db    100,101,102,103,-1,-1     ; F7 - F10, NL, SL
3730 00001837 778D848E738F                    db    119,141,132,142,115,143   ; Home, Up, PgUp, -, Left, Pad5
3731 0000183D 749075917692                    db    116,144,117,145,118,146 ; Right, +, End, Down, PgDn, Ins
3732 00001843 93FFFFFF898A                    db    147,-1,-1,-1,137,138      ; Del, SysReq, Undef, WT, F11, F12
3733
3734                                    ;-----      TABLES FOR LOWER CASE ----------
3735 00001849 1B3132333435363738-       K10:  db    27,'1234567890-=',8,9
3736 00001852 39302D3D0809
3737 00001858 71776572747975696F-             db    'qwertyuiop[]',13,-1,'asdfghjkl;',39
3738 00001861 705B5D0DFF61736466-
3739 0000186A 67686A6B6C3B27
3740 00001871 60FF5C7A786376626E-             db    96,-1,92,'zxcvbnm,./',-1,'*',-1,' ',-1
3741 0000187A 6D2C2E2FFF2AFF20FF
3742                                    ;-----      LC TABLE SCAN
3743 00001883 3B3C3D3E3F                      db    59,60,61,62,63            ; BASE STATE OF F1 - F10
3744 00001888 4041424344                      db    64,65,66,67,68
3745 0000188D FFFF                            db    -1,-1               ; NL, SL
```

```
3746
3747                                    ;-----      KEYPAD TABLE
3748 0000188F 474849FF4BFF              K15:  db    71,72,73,-1,75,-1   ; BASE STATE OF KEYPAD KEYS
3749 00001895 4DFF4F50515253                  db    77,-1,79,80,81,82,83
3750 0000189C FFFF5C8586                      db    -1,-1,92,133,134    ; SysRq, Undef, WT, F11, F12
3751
3752                                    ;-----      TABLES FOR UPPER CASE ----------
3753 000018A1 1B21402324255E262A-       K11:  db    27,'!@#$%',94,'&*()_+',8,0
3754 000018AA 28295F2B0800
3755 000018B0 51574552545955494F-             db    'QWERTYUIOP{}',13,-1,'ASDFGHJKL:"'
3756 000018B9 507B7D0DFF41534446-
3757 000018C2 47484A4B4C3A22
3758 000018C9 7EFF7C5A584356424E-             db    126,-1,'|ZXCVBNM<>?',-1,'*',-1,' ',-1
3759 000018D2 4D3C3E3FFF2AFF20FF
3760                                    ;-----      UC TABLE SCAN
3761 000018DB 5455565758                K12:  db    84,85,86,87,88            ; SHIFTED STATE OF F1 - F10
3762 000018E0 595A5B5C5D                      db    89,90,91,92,93
3763 000018E5 FFFF                            db    -1,-1               ; NL, SL
3764
3765                                    ;-----      NUM STATE TABLE
3766 000018E7 3738392D3435362B31-       K14:  db    '789-456+1230.'           ; NUMLOCK STATE OF KEYPAD KEYS
3767 000018F0 3233302E
3768                                          ;
3769 000018F4 FFFF7C8788                      db    -1,-1,124,135,136   ; SysRq, Undef, WT, F11, F12
3770
3771 000018F9 90<rept>                  Align 4
3772                                    ;----------------------------------------
3773                                    ;     VIDEO DISPLAY DATA AREA          ;
3774                                    ;----------------------------------------
3775 000018FC 03                        CRT_MODE     db     3      ; CURRENT DISPLAY MODE (TYPE)
3776 000018FD 29                        CRT_MODE_SET db     29h    ; CURRENT SETTING OF THE 3X8 REGISTER
3777                                                               ; (29h default setting for video mode 3)
3778                                                               ; Mode Select register Bits
3779                                                               ;   BIT 0 - 80x25 (1), 40x25 (0)
3780                                                               ;   BIT 1 - ALPHA (0), 320x200 GRAPHICS (1)
3781                                                               ;   BIT 2 - COLOR (0), BW (1)
3782                                                               ;   BIT 3 - Video Sig. ENABLE (1), DISABLE (0)
3783                                                               ;   BIT 4 - 640x200 B&W Graphics Mode (1)
3784                                                               ;   BIT 5 - ALPHA mode BLINKING (1)
3785                                                               ;   BIT 6, 7 - Not Used
3786
3787                                    ; Mode 0 - 2Ch = 101100b  ; 40x25 text, 16 gray colors
3788                                    ; Mode 1 - 28h = 101000b  ; 40x25 text, 16 fore colors, 8 back colors
3789                                    ; Mode 2 - 2Dh = 101101b  ; 80x25 text, 16 gray colors
3790                                    ; MODE 3 - 29h = 101001b  ; 80x25 text, 16 fore color, 8 back color
3791                                    ; Mode 4 - 2Ah = 101010b  ; 320x200 graphics, 4 colors
3792                                    ; Mode 5 - 2Eh = 101110b  ; 320x200 graphics, 4 gray colors
3793                                    ; Mode 6 - 1Eh = 011110b  ; 640x200 graphics, 2 colors
3794                                    ; Mode 7 - 29h = 101001b  ; 80x25 text, black & white colors
```

```
                                ; Mode & 37h = Video signal OFF

                                ; 26/08/2014
                                ; Retro UNIX 8086 v1 - UNIX.ASM (03/03/2014)
                                ; Derived from IBM "pc-at"
                                ; rombios source code (06/10/1985)
                                ; 'dseg.inc'

                                ;--------------------------------------;
                                ;     SYSTEM DATA AREA          ;
                                ;--------------------------------------
000018FE 00                     BIOS_BREAK   db    0            ; BIT 7=1 IF BREAK KEY HAS BEEN PRESSED

                                ;--------------------------------------
                                ;     KEYBOARD DATA AREAS       ;
                                ;--------------------------------------

000018FF 00                     KB_FLAG           db    0            ; KEYBOARD SHIFT STATE AND STATUS FLAGS
00001900 00                     KB_FLAG_1   db    0            ; SECOND BYTE OF KEYBOARD STATUS
00001901 00                     KB_FLAG_2   db    0            ; KEYBOARD LED FLAGS
00001902 00                     KB_FLAG_3   db    0            ; KEYBOARD MODE STATE AND TYPE FLAGS
00001903 00                     ALT_INPUT   db    0            ; STORAGE FOR ALTERNATE KEY PAD ENTRY
00001904 [14190000]             BUFFER_START dd   KB_BUFFER    ; OFFSET OF KEYBOARD BUFFER START
00001908 [34190000]             BUFFER_END  dd    KB_BUFFER + 32      ; OFFSET OF END OF BUFFER
0000190C [14190000]             BUFFER_HEAD dd    KB_BUFFER    ; POINTER TO HEAD OF KEYBOARD BUFFER
00001910 [14190000]             BUFFER_TAIL dd    KB_BUFFER    ; POINTER TO TAIL OF KEYBOARD BUFFER
                                ; ------    HEAD = TAIL  INDICATES THAT THE BUFFER IS EMPTY
00001914 0000<rept>             KB_BUFFER   times 16 dw 0            ; ROOM FOR 16 SCAN CODE ENTRIES

                                ; 28/08/2014
00001934 00000000               error_code: dd 0
                                ; 29/08/2014
00001938 00000000               FaultOffset: dd 0

                                ; 02/09/2014
                                ; 30/08/2014
                                ; VIDEO FUNCTIONS
                                ; (write_tty - Retro UNIX 8086 v1 - U9.ASM, 01/02/2014)

                                write_tty:
                                      ; 02/09/2014
                                      ; 30/08/2014 (Retro UNIX 386 v1 - beginning)
                                      ; 01/02/2014 (Retro UNIX 8086 v1 - last update)
                                      ; 03/12/2013 (Retro UNIX 8086 v1 - beginning)
                                      ; (Modified registers: EAX, EBX, ECX, EDX, ESI, EDI)
                                      ;
                                      ; INPUT -> AH = Color (Forecolor, Backcolor)
                                      ;        AL = Character to be written
```

```
                                        ;        EBX = Video Page (0 to 7)
                                        ;        (BH = 0 --> Video Mode 3)
                                
                                RVRT  equ    00001000b    ; VIDEO VERTICAL RETRACE BIT
                                RHRZ  equ    00000001b    ; VIDEO HORIZONTAL RETRACE BIT
                                
                                ; Derived from "WRITE_TTY" procedure of IBM "pc-at" rombios source code
                                ; (06/10/1985), 'video.asm', INT 10H, VIDEO_IO
                                ;
                                ; 06/10/85  VIDEO DISPLAY BIOS
                                ;
                                ;--- WRITE_TTY ----------------------------------------------------------------
                                ;                                                                       :
                                ;   THIS INTERFACE PROVIDES A TELETYPE LIKE INTERFACE TO THE                 :
                                ;   VIDEO CARDS. THE INPUT CHARACTER IS WRITTEN TO THE CURRENT               :
                                ;   CURSOR POSITION, AND THE CURSOR IS MOVED TO THE NEXT POSITION.           :
                                ;   IF THE CURSOR LEAVES THE LAST COLUMN OF THE FIELD, THE COLUMN            :
                                ;   IS SET TO ZERO, AND THE ROW VALUE IS INCREMENTED. IF THE ROW       :
                                ;   ROW VALUE LEAVES THE FIELD, THE CURSOR IS PLACED ON THE LAST ROW,        :
                                ;   FIRST COLUMN, AND THE ENTIRE SCREEN IS SCROLLED UP ONE LINE.       :
                                ;   WHEN THE SCREEN IS SCROLLED UP, THE ATTRIBUTE FOR FILLING THE            :
                                ;   NEWLY BLANKED LINE IS READ FROM THE CURSOR POSITION ON THE PREVIOUS      :
                                ;   LINE BEFORE THE SCROLL, IN CHARACTER MODE. IN GRAPHICS MODE,       :
                                ;   THE 0 COLOR IS USED.                                                  :
                                ;   ENTRY --                                                              :
                                ;     (AH) = CURRENT CRT MODE                                             :
                                ;     (AL) = CHARACTER TO BE WRITTEN                                      :
                                ;         NOTE THAT BACK SPACE, CARRIAGE RETURN, BELL AND LINE FEED ARE :
                                ;         HANDLED AS COMMANDS RATHER THAN AS DISPLAY GRAPHICS CHARACTERS      :
                                ;     (BL) = FOREGROUND COLOR FOR CHAR WRITE IF CURRENTLY IN A GRAPHICS MODE  :
                                ;   EXIT --                                                               :
                                ;     ALL REGISTERS SAVED                                                 :
                                ;-------------------------------------------------------------------------------
                                
0000193C FA                             cli
                                        ;
                                        ; READ CURSOR (04/12/2013)
                                        ; Retro UNIX 386 v1 Modifications: 30/08/2014
0000193D 08FF                           or     bh, bh
0000193F 0F85B5000000                   jnz    beeper
                                        ; 01/09/2014
00001945 803D[FC180000]03               cmp    byte [CRT_MODE], 3
0000194C 7405                           je     short m3
                                        ;
0000194E E895020000                     call   set_mode
                                m3:
00001953 31F6                           xor    esi, esi
00001955 6689DE                         mov    si, bx
00001958 66D1E6                         shl    si, 1
```

```
0000195B 81C6[3C490000]             add   esi, cursor_posn
00001961 668B16                     mov   dx, word [esi]
                                    ;
                                    ; dx now has the current cursor position
                                    ;
00001964 3C0D                       cmp   al, 0Dh            ; is it carriage return or control character
00001966 7654                       jbe   short u8
                                    ;
                                    ; write the char to the screen
                                u0:
                                    ; ah = attribute/color
                                    ; al = character
                                    ; bl = video page number (0 to 7)
                                    ; bh = 0
                                    ;
00001968 E84A020000                 call  write_c_current
                                    ;
                                    ; position the cursor for next char
0000196D FEC2                       inc   dl           ; next column
                                    ;cmp  dl, byte [CRT_COLS]
0000196F 80FA50                     cmp   dl, 80       ; test for column overflow
00001972 0F85F2000000                 jne     set_cpos
00001978 B200                       mov   dl, 0        ; column = 0
                                u10:                   ; (line feed found)
0000197A 80FE18                     cmp   dh, 25-1     ; check for last row
0000197D 7236                       jb    short u6
                                    ;
                                    ; scroll required
                                u1:
                                    ; SET CURSOR POSITION (04/12/2013)
0000197F E8E6000000                 call  set_cpos
                                    ;
                                    ; determine value to fill with during scroll
                                u2:
                                    ; READ_AC_CURRENT         :
                                    ;   THIS ROUTINE READS THE ATTRIBUTE AND CHARACTER
                                    ;    AT THE CURRENT CURSOR POSITION
                                    ;
                                    ; INPUT
                                    ;     (AH) = CURRENT CRT MODE
                                    ;     (BH) = DISPLAY PAGE ( ALPHA MODES ONLY )
                                    ;     (DS) = DATA SEGMENT
                                    ;     (ES) = REGEN SEGMENT
                                    ; OUTPUT
                                    ;     (AL) = CHARACTER READ
                                    ;     (AH) = ATTRIBUTE READ
                                    ;
                                    ; mov  ah, byte [CRT_MODE] ; move current mode into ah
                                    ;
```

```
3942                                            ; bl = video page number
3943                                            ;
3944 00001984 E83F010000                       call  find_position; get regen location and port address
3945                                            ; dx = status port
3946                                            ; esi = cursor location/address
3947                                    p11:
3948 00001989 FB                               sti                ; enable interrupts
3949 0000198A 90                               nop                ; allow for small interupts window
3950 0000198B FA                               cli                ; blocks interrupts for single loop
3951 0000198C EC                               in     al, dx      ; get status from adapter
3952 0000198D A801                             test   al, RHRZ    ; is horizontal retrace low
3953 0000198F 75F8                             jnz    short p11   ; wait until it is
3954                                    p12:                       ; now wait for either retrace high
3955 00001991 EC                               in     al, dx      ; get status
3956 00001992 A809                             test   al, RVRT+RHRZ; is horizontal or vertical retrace high
3957 00001994 74FB                             jz     short p12   ; wait until either is active
3958                                    p13:
3959 00001996 81C600800B00                     add    esi, 0B8000h ; 30/08/2014 (Retro UNIX 386 v1)
3960 0000199C 668B06                           mov    ax, word [esi]     ; get the character and attribute
3961                                            ;
3962                                            ; al = character, ah = attribute
3963                                            ;
3964 0000199F FB                               sti
3965                                            ; bl = video page number
3966                                    u3:
3967                                            ;;mov  ax, 0601h    ; scroll one line
3968                                            ;;sub  cx, cx       ; upper left corner
3969                                            ;;mov  dh, 25-1     ; lower right row
3970                                            ;;;mov dl, byte [CRT_COLS]
3971                                            ;mov   dl, 80       ; lower right column
3972                                            ;;dec  dl
3973                                            ;;mov  dl, 79
3974
3975                                            ;;call scroll_up    ; 04/12/2013
3976                                            ; 02/09/2014
3977 000019A0 668B0D[4E490000]                 mov    cx, word [crt_ulc] ; Upper left corner  (0000h)
3978 000019A7 668B15[50490000]                 mov    dx, word [crt_lrc] ; Lower right corner (2479h)
3979                                            ;
3980 000019AE B001                             mov    al, 1       ; scroll 1 line up
3981                                                   ; ah = attribute
3982 000019B0 E940010000                       jmp    scroll_up
3983                                    ;u4:
3984                                            ;;int  10h         ; video-call return
3985                                                               ; scroll up the screen
3986                                                               ; tty return
3987                                    ;u5:
3988                                            ;retn              ; return to the caller
3989
3990                                    u6:                        ; set-cursor-inc
```

```
3991 000019B5 FEC6                      inc    dh           ; next row
3992                                                        ; set cursor
3993                                ;u7:
3994                                    ;;mov  ah, 02h
3995                                    ;;jmp  short u4     ; establish the new cursor
3996                                    ;call  set_cpos
3997                                    ;jmp   short u5
3998 000019B7 E9AE000000               jmp     set_cpos
3999
4000                                    ; check for control characters
4001                                u8:
4002 000019BC 7438                      je     short u9
4003 000019BE 3C0A                      cmp    al, 0Ah            ; is it a line feed (0Ah)
4004 000019C0 74B8                      je     short u10
4005 000019C2 3C07                      cmp    al, 07h      ; is it a bell
4006 000019C4 7434                      je     short u11
4007 000019C6 3C08                      cmp    al, 08h            ; is it a backspace
4008                                    ;jne   short u0
4009 000019C8 7424                      je     short bs     ; 12/12/2013
4010                                    ; 12/12/2013 (tab stop)
4011 000019CA 3C09                      cmp    al, 09h            ; is it a tab stop
4012 000019CC 759A                      jne    short u0
4013 000019CE 88D0                      mov    al, dl
4014 000019D0 6698                      cbw
4015 000019D2 B108                      mov    cl, 8
4016 000019D4 F6F1                      div    cl
4017 000019D6 28E1                      sub    cl, ah
4018                                ts:
4019                                    ; 02/09/2014
4020                                    ; 01/09/2014
4021 000019D8 B020                      mov    al, 20h
4022                                tsloop:
4023 000019DA 6651                      push   cx
4024 000019DC 6650                      push   ax
4025 000019DE 30FF                      xor    bh, bh
4026                                    ;mov   bl, [active_page]
4027 000019E0 E86EFFFFFF                call   m3
4028 000019E5 6658                      pop    ax  ; ah = attribute/color
4029 000019E7 6659                      pop    cx
4030 000019E9 FEC9                      dec    cl
4031 000019EB 75ED                      jnz    short tsloop
4032 000019ED C3                        retn
4033                                bs:
4034                                    ; back space found
4035
4036 000019EE 08D2                      or     dl, dl             ; is it already at start of line
4037                                    ;je    short u7     ; set_cursor
4038 000019F0 7478                      jz     short set_cpos
4039 000019F2 664A                      dec    dx                 ; no -- just move it back
```

```
4040                                    ;jmp   short u7
4041 000019F4 EB74                      jmp    short set_cpos
4042
4043                                    ; carriage return found
4044                                u9:
4045 000019F6 B200                      mov    dl, 0        ; move to first column
4046                                    ;jmp   short u7
4047 000019F8 EB70                      jmp    short set_cpos
4048
4049                                    ; line feed found
4050                                ;u10:
4051                                ;     cmp    dh, 25-1     ; bottom of screen
4052                                ;     jne    short u6     ; no, just set the cursor
4053                                ;       jmp     u1             ; yes, scroll the screen
4054
4055                                beeper:
4056                                    ; 30/08/2014 (Retro UNIX 386 v1)
4057                                    ; 18/01/2014
4058                                    ; 03/12/2013
4059                                    ; bell found
4060                                u11:
4061 000019FA FB                        sti
4062 000019FB 3A1D[4C490000]            cmp    bl, byte [active_page]
4063 00001A01 7551                      jne    short u12    ; Do not sound the beep
4064                                                        ; if it is not written on the active page
4065 00001A03 66B93305                  mov    cx, 1331     ; divisor for 896 hz tone
4066 00001A07 B31F                      mov    bl, 31       ; set count for 31/64 second for beep
4067                                    ;call  beep         ; sound the pod bell
4068                                    ;jmp   short u5     ; tty_return
4069                                    ;retn
4070
4071                                TIMER equ    040h              ; 8254 TIMER - BASE ADDRESS
4072                                PORT_B       equ    061h       ; PORT B READ/WRITE DIAGNOSTIC REGISTER
4073                                GATE2 equ    00000001b    ; TIMER 2 INPUT CATE CLOCK BIT
4074                                SPK2  equ    00000010b    ; SPEAKER OUTPUT DATA ENABLE BIT
4075
4076                                beep:
4077                                    ; 07/02/2015
4078                                    ; 30/08/2014 (Retro UNIX 386 v1)
4079                                    ; 18/01/2014
4080                                    ; 03/12/2013
4081                                    ;
4082                                    ; TEST4.ASM - 06/10/85  POST AND BIOS UTILITY ROUTINES
4083                                    ;
4084                                    ; ROUTINE TO SOUND THE BEEPER USING TIMER 2 FOR TONE
4085                                    ;
4086                                    ; ENTRY:
4087                                    ;    (BL) = DURATION COUNTER ( 1 FOR 1/64 SECOND )
4088                                    ;    (CX) = FREQUENCY DIVISOR (1193180/FREQUENCY) (1331 FOR 886 HZ)
```

```
                                        ; EXIT:                         :
                                        ;    (AX),(BL),(CX) MODIFIED.
                                        
00001A09 9C                             pushf  ; 18/01/2014 ; save interrupt status
00001A0A FA                             cli                 ; block interrupts during update
00001A0B B0B6                           mov    al, 10110110b; select timer 2, lsb, msb binary
00001A0D E643                           out    TIMER+3, al  ; write timer mode register
00001A0F EB00                           jmp    $+2          ; I/O delay
00001A11 88C8                           mov    al, cl       ; divisor for hz (low)
00001A13 E642                           out    TIMER+2,AL   ; write timer 2 count - lsb
00001A15 EB00                           jmp    $+2          ; I/O delay
00001A17 88E8                           mov    al, ch       ; divisor for hz (high)
00001A19 E642                           out    TIMER+2, al  ; write timer 2 count - msb
00001A1B E461                           in     al, PORT_B   ; get current setting of port
00001A1D 88C4                           mov    ah, al       ; save that setting
00001A1F 0C03                           or     al, GATE2+SPK2      ; gate timer 2 and turn speaker on
00001A21 E661                           out    PORT_B, al   ; and restore interrupt status
                                        ;popf  ; 18/01/2014
00001A23 FB                             sti
                                g7:                         ; 1/64 second per count (bl)
00001A24 B90B040000                     mov    ecx, 1035    ; delay count for 1/64 of a second
00001A29 E827000000                     call   waitf        ; go to beep delay 1/64 count
00001A2E FECB                           dec    bl           ; (bl) length count expired?
00001A30 75F2                           jnz    short g7     ; no - continue beeping speaker
                                        ;
                                        ;pushf              ; save interrupt status
00001A32 FA                             cli    ; 18/01/2014 ; block interrupts during update
00001A33 E461                           in     al, PORT_B   ; get current port value
                                          ;or      al, not (GATE2+SPK2) ; isolate current speaker bits in case
00001A35 0CFC                             or      al, ~(GATE2+SPK2)
00001A37 20C4                             and  ah, al         ; someone turned them off during beep
00001A39 88E0                           mov    al, ah       ; recover value of port
                                          ;or      al, not (GATE2+SPK2) ; force speaker data off
00001A3B 0CFC                           or     al, ~(GATE2+SPK2) ; isolate current speaker bits in case
00001A3D E661                           out    PORT_B, al   ; and stop speaker timer
                                        ;popf               ; restore interrupt flag state
00001A3F FB                             sti
00001A40 B90B040000                     mov    ecx, 1035    ; force 1/64 second delay (short)
00001A45 E80B000000                     call   waitf        ; minimum delay between all beeps
                                        ;pushf              ; save interrupt status
00001A4A FA                             cli                 ; block interrupts during update
00001A4B E461                           in     al, PORT_B   ; get current port value in case
00001A4D 2403                           and    al, GATE2+SPK2      ; someone turned them on
00001A4F 08E0                           or     al, ah       ; recover value of port_b
00001A51 E661                           out    PORT_B, al   ; restore speaker status
00001A53 9D                             popf                ; restore interrupt flag state
                                u12:
00001A54 C3                             retn
```

```
                        REFRESH_BIT equ    00010000b    ; REFRESH TEST BIT
                        
                        WAITF:
                        waitf:
                              ; 30/08/2014 (Retro UNIX 386 v1)
                              ; 03/12/2013
                              ;
                        ;     push ax                  ; save work register (ah)
                        ;waitf1:
                                                       ; use timer 1 output bits
                        ;     in     al, PORT_B   ; read current counter output status
                        ;     and    al, REFRESH_BIT     ; mask for refresh determine bit
                        ;     cmp    al, ah       ; did it just change
                        ;     je     short waitf1 ; wait for a change in output line
                        ;     ;
                        ;     mov    ah, al       ; save new lflag state
                        ;     loop   waitf1       ; decrement half cycles till count end
                        ;     ;
                        ;     pop    ax           ; restore (ah)
                        ;     retn                ; return (cx)=0
                        
                        ; 06/02/2015 (unix386.s <-- dsectrm2.s)
                        ; 17/12/2014 (dsectrm2.s)
                        ; WAITF
                        ; /// IBM PC-XT Model 286 System BIOS Source Code - Test 4 - 06/10/85 ///
                        ;
                        ;---WAITF--------------------------------------------------------------------------
                        ;     FIXED TIME WAIT ROUTINE (HARDWARE CONTROLLED - NOT PROCESSOR)
                        ; ENTRY:
                        ;     (CX) = COUNT OF 15.085737 MICROSECOND INTERVALS TO WAIT
                        ;            MEMORY REFRESH TIMER 1 OUTPUT USED AS REFERENCE
                        ; EXIT:
                        ;                   AFTER (CX) TIME COUNT (PLUS OR MINUS 16 MICROSECONDS)
                        ;     (CX) = 0
                        ;-----------------------------------------------------------------------------------
                        
                        ; Refresh period: 30 micro seconds (15-80 us)
                        ; (16/12/2014 - AWARDBIOS 1999 - ATORGS.ASM, WAIT_REFRESH)
                        
                        ;WAITF:                                ; DELAY FOR (CX)*15.085737 US
00001A55 6650                 PUSH   AX                ; SAVE WORK REGISTER (AH)
                              ; 16/12/2014
                              ;shr   cx, 1             ; convert to count of 30 micro seconds
00001A57 D1E9                 shr    ecx, 1 ; 21/02/2015
                        ;17/12/2014
                        ;WAITF1:
                        ;     IN     AL, PORT_B   ;061h  ; READ CURRENT COUNTER OUTPUT STATUS
                        ;     AND    AL, REFRESH_BIT     ;00010000b ; MASK FOR REFRESH DETERMINE BIT
                        ;     CMP    AL, AH              ; DID IT JUST CHANGE
```

```
4187                                    ;       JE      short WAITF1        ; WAIT FOR A CHANGE IN OUTPUT LINE
4188                                    ;       MOV     AH, AL              ; SAVE NEW FLAG STATE
4189                                    ;       LOOP    WAITF1              ; DECREMENT HALF CYCLES TILL COUNT END
4190                                            ;
4191                                            ; 17/12/2014
4192                                            ;
4193                                            ; Modification from 'WAIT_REFRESH' procedure of AWARD BIOS - 1999
4194                                            ;
4195                                    ;WAIT_REFRESH:  Uses port 61, bit 4 to have CPU speed independent waiting.
4196                                    ;       INPUT:  CX = number of refresh periods to wait
4197                                    ;                       (refresh periods = 1 per 30 microseconds on most machines)
4198                                    WR_STATE_0:
4199 00001A59 E461                             IN      AL,PORT_B           ; IN AL,SYS1
4200 00001A5B A810                             TEST    AL,010H
4201 00001A5D 74FA                             JZ      SHORT WR_STATE_0
4202                                    WR_STATE_1:
4203 00001A5F E461                             IN      AL,PORT_B           ; IN AL,SYS1
4204 00001A61 A810                             TEST    AL,010H
4205 00001A63 75FA                             JNZ     SHORT WR_STATE_1
4206 00001A65 E2F2                               LOOP    WR_STATE_0
4207                                           ;
4208 00001A67 6658                             POP     AX                  ; RESTORE (AH)
4209 00001A69 C3                               RETn                        ; (CX) = 0
4210
4211                                    set_cpos:
4212                                           ; 01/09/2014
4213                                           ; 30/08/2014 (Retro UNIX 386 v1 - beginning)
4214                                           ;
4215                                           ; 12/12/2013 (Retro UNIX 8086 v1 - last update)
4216                                           ; 04/12/2013 (Retro UNIX 8086 v1 - beginning)
4217                                           ;
4218                                           ; VIDEO.ASM - 06/10/85  VIDEO DISPLAY BIOS
4219                                           ;
4220                                           ; SET_CPOS
4221                                           ;       THIS ROUTINE SETS THE CURRENT CURSOR POSITION TO THE
4222                                           ;       NEW X-Y VALUES PASSED
4223                                           ; INPUT
4224                                           ;       DX - ROW,COLUMN OF NEW CURSOR
4225                                           ;       BH - DISPLAY PAGE OF CURSOR
4226                                           ; OUTPUT
4227                                           ;       CURSOR IS SET AT 6845 IF DISPLAY PAGE IS CURRENT DISPLAY
4228                                           ;
4229 00001A6A 31C0                               xor     eax, eax
4230 00001A6C 88D8                               mov     al, bl  ; BL = video page number
4231 00001A6E D0E0                               shl     al, 1   ; word offset
4232 00001A70 BE[3C490000]                     mov     esi, cursor_posn
4233 00001A75 01C6                               add     esi, eax
4234 00001A77 668916                           mov     word [esi], dx ; save the pointer
4235 00001A7A 381D[4C490000]                   cmp     byte [active_page], bl
```

```
4236 00001A80 7532                      jne    short m17
4237                                    ;call  m18    ; CURSOR SET
4238                                ;m17:              ; SET_CPOS_RETURN
4239                                    ; 01/09/2014
4240                                ;   retn
4241                                           ; DX  = row/column
4242                                m18:
4243 00001A82 E832000000                call   position ; determine location in regen buffer
4244 00001A87 668B0D[3A490000]          mov    cx, word [CRT_START]
4245 00001A8E 6601C1                    add    cx, ax  ; add char position in regen buffer
4246                                                    ; to the start address (offset) for this page
4247 00001A91 66D1E9                    shr    cx, 1  ; divide by 2 for char only count
4248 00001A94 B40E                      mov    ah, 14 ; register number for cursor
4249                                    ;call  m16    ; output value to the 6845
4250                                    ;retn
4251
4252                                    ;----- THIS ROUTINE OUTPUTS THE CX REGISTER
4253                                    ;      TO THE 6845 REGISTERS NAMED IN (AH)
4254                                m16:
4255 00001A96 FA                        cli
4256                                    ;mov   dx, word [addr_6845] ; address register
4257 00001A97 66BAD403                  mov    dx, 03D4h ; I/O address of color card
4258 00001A9B 88E0                      mov    al, ah ; get value
4259 00001A9D EE                        out    dx, al ; register set
4260 00001A9E 6642                      inc    dx     ; data register
4261 00001AA0 EB00                      jmp    $+2    ; i/o delay
4262 00001AA2 88E8                      mov    al, ch ; data
4263 00001AA4 EE                        out    dx, al
4264 00001AA5 664A                      dec    dx
4265 00001AA7 88E0                      mov    al, ah
4266 00001AA9 FEC0                      inc    al     ; point to other data register
4267 00001AAB EE                        out    dx, al ; set for second register
4268 00001AAC 6642                      inc    dx
4269 00001AAE EB00                      jmp    $+2    ; i/o delay
4270 00001AB0 88C8                      mov    al, cl ; second data value
4271 00001AB2 EE                        out    dx, al
4272 00001AB3 FB                        sti
4273                                m17:
4274 00001AB4 C3                        retn
4275
4276
4277                                set_ctype:
4278                                    ; 02/09/2014 (Retro UNIX 386 v1)
4279                                    ;
4280                                    ; VIDEO.ASM - 06/10/85  VIDEO DISPLAY BIOS
4281
4282                                ;   CH) = BITS 4-0 = START LINE FOR CURSOR
4283                                ;     ** HARDWARE WILL ALWAYS CAUSE BLINK
4284                                ;     ** SETTING BIT 5 OR 6 WILL CAUSE ERRATIC BLINKING
```

```
4285                                    ;          OR NO CURSOR AT ALL
4286                                    ;     (CL) = BITS 4-0 = END LINE FOR CURSOR
4287                                    
4288                                    ;-----------------------------------------------
4289                                    ; SET_CTYPE
4290                                    ;     THIS ROUTINE SETS THE CURSOR VALUE
4291                                    ; INPUT
4292                                    ;     (CX) HAS CURSOR VALUE CH-START LINE, CL-STOP LINE
4293                                    ; OUTPUT
4294                                    ;     NONE
4295                                    ;-----------------------------------------------
4296                                    
4297 00001AB5 B40A                     	mov    ah, 10 ; 6845 register for cursor set
4298                                    	;mov   word [CURSOR_MODE], cx ; save in data area
4299                                    	;call  m16    ; output cx register
4300                                    	;retn
4301 00001AB7 EBDD                     	jmp    m16
4302                                    
4303                                    
4304                                    position:
4305                                    	; 02/09/2014
4306                                    	; 30/08/2014 (Retro UNIX 386 v1)
4307                                    	; 04/12/2013 (Retro UNIX 8086 v1)
4308                                    	;
4309                                    	; VIDEO.ASM - 06/10/85  VIDEO DISPLAY BIOS
4310                                    	;
4311                                    	; POSITION
4312                                    	;     THIS SERVICE ROUTINE CALCULATES THE REGEN BUFFER ADDRESS
4313                                    	;     OF A CHARACTER IN THE ALPHA MODE
4314                                    	; INPUT
4315                                    	;     AX = ROW, COLUMN POSITION
4316                                    	; OUTPUT
4317                                    	;     AX = OFFSET OF CHAR POSITION IN REGEN BUFFER
4318                                    
4319                                    	        ; DX = ROW, COLUMN POSITION
4320 00001AB9 31C0                     	xor    eax, eax ; 02/09/2014
4321                                    	;mov   al, byte [CRT_COLS]
4322 00001ABB B050                     	mov    al, 80   ; determine bytes to row
4323 00001ABD F6E6                     	mul    dh ;    row value
4324 00001ABF 30F6                     	xor    dh, dh   ; 0
4325 00001AC1 6601D0                   	add    ax, dx  ; add column value to the result
4326 00001AC4 66D1E0                   	shl    ax, 1  ; * 2 for attribute bytes
4327                                    	        ; EAX = AX = OFFSET OF CHAR POSITION IN REGEN BUFFER
4328 00001AC7 C3                       	retn
4329                                    
4330                                    find_position:
4331                                    	; 07/09/2014
4332                                    	; 02/09/2014
4333                                    	; 30/08/2014 (Retro UNIX 386 v1)
```

```
4334                                    ; VIDEO.ASM - 06/10/85  VIDEO DISPLAY BIOS
4335 00001AC8 31C9                     xor    ecx, ecx
4336 00001ACA 88D9                     mov    cl, bl ; video page number
4337 00001ACC 89CE                     mov    esi, ecx
4338 00001ACE 66D1E6                   shl    si, 1
4339 00001AD1 668B96[3C490000]         mov    dx, word [esi + cursor_posn]
4340 00001AD8 740A                     jz     short p21
4341 00001ADA 6631F6                   xor    si, si
4342                                p20:
4343                                    ;add   esi, word [CRT_LEN]
4344 00001ADD 6681C6A00F                add    si, 80*25*2 ; add length of buffer for one page
4345 00001AE2 E2F9                      loop   p20
4346                                p21:
4347 00001AE4 6621D2                   and    dx, dx
4348 00001AE7 7407                     jz     short p22
4349 00001AE9 E8CBFFFFFF               call   position ; determine location in regen in page
4350 00001AEE 01C6                     add    esi, eax ; add location to start of regen page
4351                                p22:
4352                                    ;mov   dx, word [addr_6845] ; get base address of active display

4353                                    ;mov   dx, 03D4h ; I/O address of color card
4354                                    ;add   dx, 6  ; point at status port
4355 00001AF0 66BADA03                  mov    dx, 03DAh ; status port
4356                                    ; cx = 0
4357 00001AF4 C3                        retn
4358
4359                                scroll_up:
4360                                    ; 07/09/2014
4361                                    ; 02/09/2014
4362                                    ; 01/09/2014 (Retro UNIX 386 v1 - beginning)
4363                                    ; 04/04/2014
4364                                    ; 04/12/2013
4365                                    ;
4366                                    ; VIDEO.ASM - 06/10/85  VIDEO DISPLAY BIOS
4367                                    ;
4368                                    ; SCROLL UP
4369                                    ;      THIS ROUTINE MOVES A BLOCK OF CHARACTERS UP
4370                                    ;      ON THE SCREEN
4371                                    ; INPUT
4372                                    ;      (AH) = CURRENT CRT MODE
4373                                    ;      (AL) = NUMBER OF ROWS TO SCROLL
4374                                    ;      (CX) = ROW/COLUMN OF UPPER LEFT CORNER
4375                                    ;      (DX) = ROW/COLUMN OF LOWER RIGHT CORNER
4376                                    ;      (BH) = ATTRIBUTE TO BE USED ON BLANKED LINE
4377                                    ;      (DS) = DATA SEGMENT
4378                                    ;      (ES) = REGEN BUFFER SEGMENT
4379                                    ; OUTPUT
4380                                    ;      NONE -- THE REGEN BUFFER IS MODIFIED
4381                                    ;
```

```
4382                                    ;      bh = 0  (02/09/2014)
4383                                    ;
4384                                    ; ((ah = 3))
4385                                    ; cl = left upper column
4386                                    ; ch = left upper row
4387                                    ; dl = righ lower column
4388                                    ; dh = righ lower row
4389                                    ;
4390                                    ; al = line count
4391                                    ; ah = attribute to be used on blanked line
4392                                    ; bl = video page number (0 to 7)
4393                                    ;
4394
4395                                    ; Test Line Count
4396 00001AF5 08C0                      or     al, al
4397 00001AF7 740C                      jz     short al_set
4398 00001AF9 88F7                      mov    bh, dh ; subtract lower row from upper row
4399 00001AFB 28EF                      sub    bh, ch
4400 00001AFD FEC7                      inc    bh     ; adjust difference by 1
4401 00001AFF 38C7                      cmp    bh, al       ; line count = amount of rows in window?
4402 00001B01 7502                      jne    short al_set ; if not the we're all set
4403 00001B03 30C0                      xor    al, al ; otherwise set al to zero
4404                                 al_set:
4405 00001B05 30FF                            xor    bh, bh ; 0
4406 00001B07 6650                            push   ax
4407                                          ;mov   esi, [crt_base]
4408 00001B09 BE00800B00                        mov     esi, 0B8000h
4409 00001B0E 3A1D[4C490000]                    cmp     bl, [active_page]
4410 00001B14 750B                            jne    short n0
4411                                          ;
4412 00001B16 66A1[3A490000]                    mov     ax, [CRT_START]
4413 00001B1C 6601C6                            add     si, ax
4414 00001B1F EB0F                              jmp     short n1
4415                                 n0:
4416 00001B21 20DB                              and     bl, bl
4417 00001B23 740B                            jz     short n1
4418 00001B25 88D8                            mov    al, bl
4419                                 n0x:
4420                                            ;add    si, word [CRT_LEN]
4421                                            ;add    esi, 80*25*2
4422 00001B27 6681C6A00F                        add     si, 80*25*2
4423 00001B2C FEC8                              dec  al
4424 00001B2E 75F7                            jnz    short n0x
4425                                 n1:
4426                                            ;Scroll position
4427 00001B30 6652                            push   dx
4428 00001B32 6689CA                          mov    dx, cx ; now, upper left position in DX
4429 00001B35 E87FFFFFFF                      call   position
4430 00001B3A 01C6                            add    esi, eax
```

```
4431 00001B3C 89F7                      mov   edi, esi
4432 00001B3E 665A                      pop   dx    ; lower right position in DX
4433 00001B40 6629CA                    sub   dx, cx
4434 00001B43 FEC6                      inc   dh    ; dh = #rows
4435 00001B45 FEC2                      inc   dl    ; dl = #cols in block
4436 00001B47 6658                      pop   ax    ; al = line count, ah = attribute
4437 00001B49 31C9                      xor   ecx, ecx
4438 00001B4B 6689C1                    mov   cx, ax
4439                                    ;mov  ah, byte [CRT_COLS]
4440 00001B4E B450                      mov   ah, 80
4441 00001B50 F6E4                      mul   ah    ; determine offset to from address
4442 00001B52 6601C0                    add   ax, ax  ; *2 for attribute byte
4443                                    ;
4444 00001B55 6650                      push  ax    ; offset
4445 00001B57 6652                      push  dx
4446                                    ;
4447                                    ; 04/04/2014
4448 00001B59 66BADA03                  mov   dx, 3DAh ; guaranteed to be color card here
4449                                n8:                    ; wait_display_enable
4450 00001B5D EC                          in    al, dx   ; get port
4451 00001B5E A808                      test  al, RVRT ; wait for vertical retrace
4452 00001B60 74FB                      jz    short n8 ; wait_display_enable
4453 00001B62 B025                      mov   al, 25h
4454 00001B64 B2D8                      mov   dl, 0D8h ; address control port
4455 00001B66 EE                        out   dx, al ; turn off video during vertical retrace
4456 00001B67 665A                      pop   dx    ; #rows, #cols
4457 00001B69 6658                            pop   ax    ; offset
4458 00001B6B 6691                      xchg  ax, cx ;
4459                                    ; ecx = offset, al = line count, ah = attribute
4460                                ;n9:
4461 00001B6D 08C0                      or    al, al
4462 00001B6F 7420                        jz    short n3
4463 00001B71 01CE                        add   esi, ecx ; from address for scroll
4464 00001B73 88F7                      mov   bh, dh  ; #rows in block
4465 00001B75 28C7                      sub   bh, al ; #rows to be moved
4466                                n2:
4467                                    ; Move rows
4468 00001B77 88D1                      mov   cl, dl ; get # of cols to move
4469 00001B79 56                        push  esi
4470 00001B7A 57                        push  edi   ; save start address
4471                                n10:
4472 00001B7B 66A5                      movsw       ; move that line on screen
4473 00001B7D FEC9                      dec   cl
4474 00001B7F 75FA                        jnz   short n10
4475 00001B81 5F                        pop   edi
4476 00001B82 5E                        pop   esi   ; recover addresses
4477                                      ;mov  cl, byte [CRT_COLS]
4478                                    ;add  cl, cl
4479                                      ;mov  ecx, 80*2
```

```
4480 00001B83 66B9A000                          mov     cx, 80*2
4481 00001B87 01CE                              add     esi, ecx  ; next line
4482 00001B89 01CF                              add     edi, ecx
4483 00001B8B FECF                            dec   bh      ; count of lines to move
4484 00001B8D 75E8                            jnz   short n2 ; row loop
4485                                          ; bh = 0
4486 00001B8F 88C6                            mov   dh, al  ; #rows
4487                                    n3:
4488                                          ; attribute in ah
4489 00001B91 B020                            mov   al, ' '       ; fill with blanks
4490                                          ; Clear rows
4491                                                  ; dh =  #rows
4492 00001B93 88D1                              mov cl, dl ; get # of cols to clear
4493 00001B95 57                                push    edi     ; save address
4494                                    n11:
4495 00001B96 66AB                              stosw           ; store fill character
4496 00001B98 FEC9                            dec   cl
4497 00001B9A 75FA                              jnz     short n11
4498 00001B9C 5F                                pop     edi     ; recover address
4499                                          ;mov  cl, byte [CRT_COLS]
4500                                          ;add  cl, cl
4501                                            ;mov    ecx, 80*2
4502 00001B9D B1A0                              mov cl, 80*2
4503 00001B9F 01CE                              add     esi, ecx  ; next line
4504 00001BA1 01CF                              add     edi, ecx
4505 00001BA3 FECE                            dec   dh
4506 00001BA5 75EA                            jnz   short n3
4507                                          ;
4508 00001BA7 3A1D[4C490000]                  cmp   bl, byte [active_page]
4509 00001BAD 7507                            jne   short n6
4510                                          ;mov  al, byte [CRT_MODE_SET] ; get the value of mode set
4511 00001BAF B029                            mov   al, 29h ; (ORGS.ASM), M7 mode set table value for mode 3
4512 00001BB1 66BAD803                        mov   dx, 03D8h ; always set color card port
4513 00001BB5 EE                              out   dx, al
4514                                    n6:
4515 00001BB6 C3                              retn
4516
4517
4518                                    write_c_current:
4519                                          ; 30/08/2014 (Retro UNIX 386 v1)
4520                                          ; 18/01/2014
4521                                          ; 04/12/2013
4522                                          ;
4523                                          ; VIDEO.ASM - 06/10/85  VIDEO DISPLAY BIOS
4524                                          ;
4525                                          ; WRITE_C_CURRENT
4526                                          ;     THIS ROUTINE WRITES THE CHARACTER AT
4527                                          ;     THE CURRENT CURSOR POSITION, ATTRIBUTE UNCHANGED
4528                                          ; INPUT
```

```
4529                                    ;       (AH) = CURRENT CRT MODE
4530                                    ;       (BH) = DISPLAY PAGE
4531                                    ;       (CX) = COUNT OF CHARACTERS TO WRITE
4532                                    ;       (AL) = CHAR TO WRITE
4533                                    ;       (DS) = DATA SEGMENT
4534                                    ;       (ES) = REGEN SEGMENT
4535                                    ; OUTPUT
4536                                    ;       DISPLAY REGEN BUFFER UPDATED
4537
4538 00001BB7 FA                        cli
4539                                    ; bl = video page
4540                                    ; al = character
4541                                    ; ah = color/attribute
4542 00001BB8 6652                      push  dx
4543 00001BBA 6650                      push  ax     ; save character & attribute/color
4544 00001BBC E807FFFFFF                call  find_position  ; get regen location and port address
4545                                    ; esi = regen location
4546                                    ; dx = status port
4547                                    ;
4548                                    ; WAIT FOR HORIZONTAL RETRACE OR VERTICAL RETRACE
4549                                    ;
4550                                p41:                ; wait for horizontal retrace is low or vertical
4551 00001BC1 FB                        sti             ; enable interrupts first
4552 00001BC2 3A1D[4C490000]              cmp     bl, byte [active_page]
4553 00001BC8 7510                      jne   short p44
4554 00001BCA FA                        cli             ; block interrupts for single loop
4555 00001BCB EC                        in    al, dx ; get status from the adapter
4556 00001BCC A808                      test  al, RVRT ; check for vertical retrace first
4557 00001BCE 7509                      jnz   short p43 ; Do fast write now if vertical retrace
4558 00001BD0 A801                      test  al, RHRZ ; is horizontal retrace low
4559 00001BD2 75ED                      jnz   short p41 ; wait until it is
4560                                p42:                ;  wait for either retrace high
4561 00001BD4 EC                        in    al, dx ; get status again
4562 00001BD5 A809                      test  al, RVRT+RHRZ ; is horizontal or vertical retrace high
4563 00001BD7 74FB                      jz    short p42 ; wait until either retrace active
4564                                p43:
4565 00001BD9 FB                        sti
4566                                p44:
4567 00001BDA 6658                      pop   ax     ; restore the character (al) & attribute (ah)
4568 00001BDC 81C600800B00              add   esi, 0B8000h ; 30/08/2014 (crt_base)
4569                                                       ; Retro UNIX 386 v1 feature only!
4570 00001BE2 668906                    mov   word [esi], ax
4571 00001BE5 665A                      pop   dx
4572 00001BE7 C3                        retn
4573
4574                                set_mode:
4575                                    ; 02/09/2014 (Retro UNIX 386 v1)
4576                                    ;
4577                                    ; VIDEO.ASM - 06/10/85  VIDEO DISPLAY BIOS
```

```
4578
4579                                    ;--------------------------------------------------
4580                                    ; SET MODE                                       :
4581                                    ;     THIS ROUTINE INITIALIZES THE ATTACHMENT TO    :
4582                                    ;     THE SELECTED MODE, THE SCREEN IS BLANKED.     :
4583                                    ; INPUT                                          :
4584                                    ;     (AL) - MODE SELECTED (RANGE 0-7)       :
4585                                    ; OUTPUT                                         :
4586                                    ;     NONE                                       :
4587                                    ;--------------------------------------------------
4588
4589 00001BE8 53                               push   ebx
4590 00001BE9 52                               push   edx
4591 00001BEA 50                                 push    eax
4592
4593                                           ;mov   dx, 03D4h    ; address or color card
4594 00001BEB B003                             mov    al, 3
4595                                    ;M8:
4596 00001BED A2[FC180000]                     mov    byte [CRT_MODE], al ; save mode in global variable
4597 00001BF2 B029                             mov    al, 29h
4598                                           ;mov   byte [CRT_MODE_SET], al ; save the mode set value
4599 00001BF4 2437                             and    al, 037h     ; video off, save high resolution bit
4600                                           ;push  dx           ; save port value
4601                                           ;add   dx, 4        ; point to control register
4602 00001BF6 66BAD803                         mov    dx, 3D8h
4603 00001BFA EE                               out    dx, al       ; reset video to off to suppress rolling
4604                                           ;pop   dx
4605                                    ;M9:
4606 00001BFB 30E4                             xor    ah, ah
4607 00001BFD BB[351C0000]                     mov    ebx, video_params ; initialization table
4608                                           ;mov   ax, word [ebx+10] ; get the cursor mode from the table
4609                                           ;xchg  ah, al
4610                                           ;mov   word [CURSOR_MODE], ax ; save cursor mode
4611 00001C02 30E4                             xor    ah, ah         ; ah is register number during loop
4612
4613                                    ;-----      LOOP THROUGH TABLE, OUTPUTTING REGISTER ADDRESS, THEN VALUE FROM TABLE
4614
4615                                    M10:               ;  initialization loop
4616 00001C04 88E0                             mov    al, ah       ; get 6845 register number
4617 00001C06 EE                               out    dx, al
4618 00001C07 6642                             inc    dx      ; point to data port
4619 00001C09 FEC4                             inc    ah     ; next register value
4620 00001C0B 8A03                             mov    al, [ebx] ; get table value
4621 00001C0D EE                               out    dx, al ; out to chip
4622 00001C0E 43                               inc    ebx    ; next in table
4623 00001C0F 664A                             dec    dx     ; back to pointer register
4624 00001C11 E2F1                             loop   M10    ; do the whole table
4625
4626                                    ;-----      FILL REGEN AREA WITH BLANK
```

```
4627                                    ;xor   ax, ax
4628                                    ;mov   word [CRT_START], ax ; start address saved in global
4629                                    ;mov   byte [ACTIVE_PAGE], al ; 0 ; (re)set page value
4630                                    ;mov   ecx, 8192 ; number of words in color card
4631                                    ; black background, light gray characeter color, space character
4632                                    ;mov   ax, 0720h ; fill char for alpha - attribute
4633                                ;M13:                ; clear buffer
4634                                    ;add   edi, 0B8000h ; [crt_base]
4635                                    ;rep   stosw  ; FILL THE REGEN BUFFER WITH BLANKS
4636
4637                                ;-----      ENABLE VIDEO AND CORRECT PORT SETTING
4638                                    ;mov   dx, 3D4h ; mov dx, word [ADDR_6845]
4639                                                    ; prepare to output to video enable port
4640                                    ;add   dx,4    ; point to the mode control gerister
4641 00001C13 66BAD803                 mov    dx, 3D8h
4642                                    ;mov   al, byte [CRT_MODE_SET] ; get the mode set value
4643 00001C17 B029                     mov    al, 29h
4644 00001C19 EE                       out    dx, al  ; set video enable port
4645
4646                                ;-----      DETERMINE NUMBER OF COLUMNS, BOTH FOR ENTIRE DISPLAY
4647                                ;-----      AND THE NUMBER TO BE USED FOR TTY INTERFACE
4648                                    ;
4649                                    ;mov byte [CRT_COLS], 80h ; initialize number of columns count
4650                                    ;
4651                                ;-----      SET CURSOR POSITIONS
4652 00001C1A 57                       push   edi
4653                                    ;mov   word [CRT_LEN], 80*25*2
4654 00001C1B 51                       push   ecx
4655 00001C1C BF[3C490000]             mov    edi, cursor_posn
4656 00001C21 B904000000               mov    ecx, 4 ; clear all cursor positions (16 bytes)
4657 00001C26 31C0                     xor    eax, eax
4658 00001C28 F3AB                     rep    stosd  ; fill with zeroes
4659 00001C2A 59                       pop    ecx
4660 00001C2B 5F                       pop    edi
4661
4662                                ;-----      SET UP OVERSCAN REGISTER
4663 00001C2C 6642                     inc    dx     ; set overscan port to a default
4664 00001C2E B030                     mov    al, 30h      ; 30H valuye for all modes except 640X200 bw
4665                                ;M14:
4666 00001C30 EE                       out    dx, al ; output the correct value to 3D9 port
4667                                    ;mov   byte [CRT_PALETTE], al ; save the value for future use
4668
4669                                ;-----      NORMAL RETURN FROM ALL VIDEO RETURNS
4670                                    ;
4671 00001C31 58                       pop    eax
4672 00001C32 5A                       pop    edx
4673 00001C33 5B                       pop    ebx
4674 00001C34 C3                       retn
4675
```

```
4676                                    video_params:
4677                                          ; 02/09/2014 (Retro UNIX 386 v1)
4678                                          ;ORGS.ASM ----- 06/10/85   COMPATIBILITY MODULE
4679                                          ; VIDEO MODE 3
4680 00001C35 71505A0A1F0619                 db     71h,50h,5Ah,0Ah,1Fh,6,19h ; SET UP FOR 80X25
4681 00001C3C 1C02070607                     db     1Ch,2,7,6,7  ; cursor start = 6, cursor stop = 7
4682 00001C41 00000000                       db     0,0,0,0
4683
4684                                    tty_sw:
4685                                          ; 07/09/2014
4686                                          ; 02/09/2014 (Retro UNIX 386 v1 - beginning)
4687                                          ;
4688                                            ;mov     byte [u.quant], 0  ; 04/03/2014
4689                                          ;
4690                                    ;act_disp_page:
4691                                          ; 04/03/2014  (act_disp_page --> tty_sw)
4692                                          ; 10/12/2013
4693                                          ; 04/12/2013
4694                                          ;
4695                                          ; VIDEO.ASM - 06/10/85  VIDEO DISPLAY BIOS
4696                                          ;
4697                                          ; ACT_DISP_PAGE
4698                                          ;     THIS ROUTINE SETS THE ACTIVE DISPLAY PAGE, ALLOWING
4699                                          ;     THE FULL USE OF THE MEMORY SET ASIDE FOR THE VIDEO ATTACHMENT
4700                                          ; INPUT
4701                                          ;     AL HAS THE NEW ACTIVE DISPLAY PAGE
4702                                          ; OUTPUT
4703                                          ;     THE 6845 IS RESET TO DISPLAY THAT PAGE
4704
4705                                          ;cli
4706
4707 00001C45 6656                           push   si ; 10/12/2013
4708                                         ;push  ebx
4709 00001C47 6651                           push   cx
4710 00001C49 6652                           push   dx
4711                                         ;
4712 00001C4B A2[4C490000]                   mov    byte [active_page], al ; save active page value ; [ptty]
4713                                         ;mov   cx, word [CRT_LEN] ; get saved length of regen buffer
4714 00001C50 66B9A00F                       mov    cx, 25*80*2
4715                                         ; 01/09/2014
4716 00001C54 31DB                           xor    ebx, ebx
4717 00001C56 88C3                           mov    bl, al
4718                                         ;
4719 00001C58 6698                           cbw    ; 07/09/2014 (ah=0)
4720 00001C5A 66F7E1                         mul    cx     ; display page times regen length
4721                                         ; 10/12/2013
4722 00001C5D 66A3[3A490000]                 mov    word [CRT_START], ax ; save start address for later
4723 00001C63 6689C1                         mov    cx, ax ; start address to cx
4724                                         ;sar   cx, 1
```

```
4725 00001C66 66D1E9                    shr    cx, 1  ; divide by 2 for 6845 handling
4726 00001C69 B40C                      mov    ah, 12 ; 6845 register for start address
4727 00001C6B E826FEFFFF                call   m16
4728                                    ;sal   bx, 1
4729                                    ; 01/09/2014
4730 00001C70 D0E3                      shl    bl, 1  ; *2 for word offset
4731 00001C72 81C3[3C490000]            add    ebx, cursor_posn
4732 00001C78 668B13                    mov    dx, word [ebx] ; get cursor for this page
4733 00001C7B E802FEFFFF                call   m18
4734                                    ;
4735 00001C80 665A                      pop    dx
4736 00001C82 6659                      pop    cx
4737                                    ;pop   ebx
4738 00001C84 665E                      pop    si ; 10/12/2013
4739                                    ;
4740                                    ;sti
4741                                    ;
4742 00001C86 C3                        retn
4743
4744
4745                                ; Write memory information
4746                                ; Temporay Code
4747                                ; 06/11/2014
4748                                ;
4749                                memory_info:
4750 00001C87 A1[24490000]                mov    eax, [memory_size] ; in pages
4751 00001C8C 50                          push   eax
4752 00001C8D C1E00C                      shl    eax, 12                ; in bytes
4753 00001C90 BB0A000000                  mov    ebx, 10
4754 00001C95 89D9                        mov    ecx, ebx        ; 10
4755 00001C97 BE[8C490000]                mov    esi, mem_total_b_str
4756 00001C9C E8D5000000                  call   bintdstr
4757 00001CA1 58                          pop    eax
4758 00001CA2 B107                        mov    cl, 7
4759 00001CA4 BE[B1490000]                mov    esi, mem_total_p_str
4760 00001CA9 E8C8000000                  call   bintdstr
4761 00001CAE A1[28490000]                mov    eax, [free_pages] ; in pages
4762 00001CB3 50                          push   eax
4763 00001CB4 C1E00C                      shl    eax, 12              ; in bytes
4764 00001CB7 B10A                        mov    cl, 10
4765 00001CB9 BE[1A4A0000]                mov    esi, free_mem_b_str
4766 00001CBE E8B3000000                  call   bintdstr
4767 00001CC3 58                          pop    eax
4768 00001CC4 B107                        mov    cl, 7
4769 00001CC6 BE[3D4A0000]                mov    esi, free_mem_p_str
4770 00001CCB E8A6000000                  call   bintdstr
4771 00001CD0 8B35[2C490000]              mov    esi, [next_page]
4772 00001CD6 89F0                        mov    eax, esi
4773 00001CD8 C1E003                      shl    eax, 3 ; 1 byte = 8 pages
```

```
4774 00001CDB 81C600001000              add   esi, MEM_ALLOC_TBL
4775 00001CE1 8B0E                      mov   ecx, [esi]
4776 00001CE3 21C9                      and   ecx, ecx
4777 00001CE5 7407                      jz    short mim1
4778                                mim0:
4779 00001CE7 D1E9                      shr   ecx, 1
4780 00001CE9 7203                      jc    short mim1
4781 00001CEB 40                        inc   eax
4782 00001CEC EBF9                      jmp   short mim0
4783                                mim1:
4784 00001CEE B907000000                mov   ecx, 7
4785 00001CF3 BE[604A0000]              mov   esi, next_mem_p_str
4786 00001CF8 E879000000                call  bintdstr
4787 00001CFD A1[38490000]              mov   eax, [mat_size]
4788 00001D02 B104                      mov   cl, 4
4789 00001D04 BE[784A0000]              mov   esi, mat_p_str
4790 00001D09 E868000000                call  bintdstr
4791 00001D0E A1[24490000]              mov   eax, [memory_size]
4792 00001D13 05FF030000                add   eax, 1023
4793 00001D18 C1E80A                    shr   eax, 10      ; Page table count
4794 00001D1B B104                      mov   cl, 4
4795 00001D1D BE[994A0000]              mov   esi, pt_c_str
4796 00001D22 E84F000000                call  bintdstr
4797 00001D27 66A1[1C490000]            mov   ax, [mem_1m_1k]
4798 00001D2D B105                      mov   cl, 5
4799 00001D2F BE[D4490000]              mov   esi, mem_1m_1k_str
4800 00001D34 E83D000000                call  bintdstr
4801 00001D39 66A1[1E490000]            mov   ax, [mem_16m_64k]
4802 00001D3F B105                      mov   cl, 5
4803 00001D41 BE[F8490000]              mov   esi, mem_16m_64k_str
4804 00001D46 E82B000000                call  bintdstr
4805                                    ;
4806 00001D4B E843000000                call  calc_free_mem
4807 00001D50 89D0                      mov   eax, edx ; free memory in pages
4808                                    ; ecx = 0
4809 00001D52 B107                      mov   cl, 7
4810 00001D54 BE[BE4A0000]              mov   esi, free_mem_c_str
4811 00001D59 E818000000                call  bintdstr
4812                                    ;
4813 00001D5E BE[62490000]              mov   esi, msg_memory_info
4814                                pmim:
4815 00001D63 AC                        lodsb
4816 00001D64 08C0                      or    al, al
4817 00001D66 740D                      jz    short pmim_ok
4818 00001D68 56                        push  esi
4819 00001D69 31DB                      xor   ebx, ebx ; 0
4820                                                 ; Video page 0 (bl=0)
4821 00001D6B B407                      mov   ah, 07h ; Black background,
4822                                                   ; light gray forecolor
```

```
4823 00001D6D E8CAFBFFFF                call   write_tty
4824 00001D72 5E                        pop    esi
4825 00001D73 EBEE                      jmp    short pmim
4826                                  pmim_ok:
4827 00001D75 C3                        retn
4828
4829                                  ; Convert binary number to decimal/numeric string
4830                                  ; 06/11/2014
4831                                  ; Temporay Code
4832                                  ;
4833
4834                                  bintdstr:
4835                                        ; EAX = binary number
4836                                        ; ESI = decimal/numeric string address
4837                                        ; EBX = divisor (10)
4838                                        ; ECX = string length (<=10)
4839 00001D76 01CE                      add    esi, ecx
4840                                  btdstr0:
4841 00001D78 4E                        dec    esi
4842 00001D79 31D2                      xor    edx, edx
4843 00001D7B F7F3                      div    ebx
4844 00001D7D 80C230                    add    dl, 30h
4845 00001D80 8816                      mov    byte [esi], dl
4846 00001D82 FEC9                      dec    cl
4847 00001D84 740C                      jz     btdstr2
4848 00001D86 09C0                      or     eax, eax
4849 00001D88 75EE                      jnz    short btdstr0
4850                                  btdstr1:
4851 00001D8A 4E                        dec    esi
4852 00001D8B C60620                      mov     byte [esi], 20h ; blank space
4853 00001D8E FEC9                      dec    cl
4854 00001D90 75F8                      jnz    short btdstr1
4855                                  btdstr2:
4856 00001D92 C3                        retn
4857
4858                                  ; Calculate free memory pages on M.A.T.
4859                                  ; 06/11/2014
4860                                  ; Temporary Code
4861                                  ;
4862
4863                                  calc_free_mem:
4864 00001D93 31D2                      xor    edx, edx
4865                                     ;xor   ecx, ecx
4866 00001D95 668B0D[38490000]          mov    cx, [mat_size] ; in pages
4867 00001D9C C1E10A                    shl    ecx, 10      ; 1024 dwords per page
4868 00001D9F BE00001000                mov    esi, MEM_ALLOC_TBL
4869                                  cfm0:
4870 00001DA4 AD                        lodsd
4871 00001DA5 51                        push   ecx
```

```
4872 00001DA6 B920000000                mov   ecx, 32
4873                                cfm1:
4874 00001DAB D1E8                      shr   eax, 1
4875 00001DAD 7301                      jnc   short cfm2
4876 00001DAF 42                        inc   edx
4877                                cfm2:
4878 00001DB0 E2F9                      loop  cfm1
4879 00001DB2 59                        pop   ecx
4880 00001DB3 E2EF                      loop  cfm0
4881 00001DB5 C3                        retn
4882
4883                                ; 06/02/2015
4884                                diskette_io:
4885 00001DB6 9C                        pushfd
4886 00001DB7 0E                        push  cs
4887 00001DB8 E892000000                call  DISKETTE_IO_1
4888 00001DBD C3                        retn
4889
4890                                ;;;;;; DISKETTE I/O ;;;;;;;;;;;;;;;;;;;; 06/02/2015 ;;;
4891                                ;////////////////////////////////////////////////////
4892
4893                                ; DISKETTE I/O - Erdogan Tan (Retro UNIX 386 v1 project)
4894                                ; 20/02/2015
4895                                ; 06/02/2015 (unix386.s)
4896                                ; 16/12/2014 - 02/01/2015 (dsectrm2.s)
4897                                ;
4898                                ; Code (DELAY) modifications - AWARD BIOS 1999 (ADISK.EQU, COMMON.MAC)
4899                                ;
4900                                ; ADISK.EQU
4901
4902                                ;----- Wait control constants
4903
4904                                ;amount of time to wait while RESET is active.
4905
4906                                WAITCPU_RESET_ON   EQU   21          ;Reset on must last at least 14us
4907                                                                     ;at 250 KBS xfer rate.
4908                                                                     ;see INTEL MCS, 1985, pg. 5-456
4909
4910                                WAITCPU_FOR_STATUS EQU   100         ;allow 30 microseconds for
4911                                                                     ;status register to become valid
4912                                                                     ;before re-reading.
4913
4914                                ;After sending a byte to NEC, status register may remain
4915                                ;incorrectly set for 24 us.
4916
4917                                WAITCPU_RQM_LOW          EQU   24          ;number of loops to check for
4918                                                                     ;RQM low.
4919
4920                                ; COMMON.MAC
```

```
4921                                    ;
4922                                    ;     Timing macros
4923                                    ;
4924
4925                                    %macro              SIODELAY 0                  ; SHORT IODELAY
4926                                                jmp short $+2
4927                                    %endmacro
4928
4929                                    %macro              IODELAY  0                  ; NORMAL IODELAY
4930                                                jmp short $+2
4931                                                jmp short $+2
4932                                    %endmacro
4933
4934                                    %macro              NEWIODELAY 0
4935                                                out    0ebh,al
4936                                    %endmacro
4937
4938                                    ; (According to) AWARD BIOS 1999 - ATORGS.ASM (dw -> equ, db -> equ)
4939                                    ;;; WAIT_FOR_MEM
4940                                    WAIT_FDU_INT_LO           equ    017798       ; 2.5 secs in 30 micro units.
4941                                    WAIT_FDU_INT_HI           equ    1
4942                                    ;;; WAIT_FOR_PORT
4943                                    WAIT_FDU_SEND_LO   equ    16667        ; .5 secons in 30 us units.
4944                                    WAIT_FDU_SEND_HI   equ    0
4945                                    ;Time to wait while waiting for each byte of NEC results = .5
4946                                    ;seconds.  .5 seconds = 500,000 micros.  500,000/30 = 16,667.
4947                                    WAIT_FDU_RESULTS_LO       equ    16667        ; .5 seconds in 30 micro units.
4948                                    WAIT_FDU_RESULTS_HI       equ    0
4949                                    ;;; WAIT_REFRESH
4950                                    ;amount of time to wait for head settle, per unit in parameter
4951                                    ;table = 1 ms.
4952                                    WAIT_FDU_HEAD_SETTLE      equ    33           ; 1 ms in 30 micro units.
4953
4954
4955                                    ; ///////////////// DISKETTE I/O ///////////////
4956
4957                                    ; 11/12/2014 (copy from IBM PC-XT Model 286 BIOS - POSTEQU.INC)
4958
4959                                    ;--------------------------------------
4960                                    ;     EQUATES USED BY POST AND BIOS    :
4961                                    ;--------------------------------------
4962
4963                                    ;--------- 8042 KEYBOARD INTERFACE AND DIAGNOSTIC CONTROL REGISTERS ------------
4964                                    ;PORT_A            EQU    060H         ; 8042 KEYBOARD SCAN CODE/CONTROL PORT
4965                                    ;PORT_B            EQU    061H         ; PORT B READ/WRITE DIAGNOSTIC REGISTER
4966                                    ;REFRESH_BIT EQU    00010000B    ; REFRESH TEST BIT
4967
4968                                    ;--------------------------------------
4969                                    ;     CMOS EQUATES FOR THIS SYSTEM     :
```

```
4970                    ;--------------------------------------------------------------------------------
4971                    ;CMOS_PORT   EQU    070H         ; I/O ADDRESS OF CMOS ADDRESS PORT
4972                    ;CMOS_DATA   EQU    071H         ; I/O ADDRESS OF CMOS DATA PORT
4973                    ;NMI         EQU    10000000B    ; DISABLE NMI INTERRUPTS MASK -
4974                                                     ;  HIGH BIT OF CMOS LOCATION ADDRESS
4975
4976                    ;---------- CMOS TABLE LOCATION ADDRESS'S ## ----------------------------------
4977                    CMOS_DISKETTE      EQU    010H         ; DISKETTE DRIVE TYPE BYTE      ;
4978                    ;           EQU    011H         ; - RESERVED                    ;C
4979                    CMOS_DISK    EQU    012H         ; FIXED DISK TYPE BYTE                ;H
4980                    ;           EQU    013H         ; - RESERVED                    ;E
4981                    CMOS_EQUIP   EQU    014H         ; EQUIPMENT WORD LOW BYTE       ;C
4982
4983                    ;---------- DISKETTE EQUATES ---------------------------------------------------
4984                    INT_FLAG     EQU    10000000B    ; INTERRUPT OCCURRENCE FLAG
4985                    DSK_CHG      EQU    10000000B    ; DISKETTE CHANGE FLAG MASK BIT
4986                    DETERMINED   EQU    00010000B    ; SET STATE DETERMINED IN STATE BITS
4987                    HOME         EQU    00010000B    ; TRACK 0 MASK
4988                    SENSE_DRV_ST EQU    00000100B    ; SENSE DRIVE STATUS COMMAND
4989                    TRK_SLAP     EQU    030H         ; CRASH STOP (48 TPI DRIVES)
4990                    QUIET_SEEK   EQU    00AH         ; SEEK TO TRACK 10
4991                    ;MAX_DRV     EQU    2            ; MAX NUMBER OF DRIVES
4992                    HD12_SETTLE  EQU    15           ; 1.2 M HEAD SETTLE TIME
4993                    HD320_SETTLE EQU    20           ; 320 K HEAD SETTLE TIME
4994                    MOTOR_WAIT   EQU    37           ; 2 SECONDS OF COUNTS FOR MOTOR TURN OFF
4995
4996                    ;---------- DISKETTE ERRORS ----------------------------------------------------
4997                    ;TIME_OUT    EQU    080H         ; ATTACHMENT FAILED TO RESPOND
4998                    ;BAD_SEEK    EQU    040H         ; SEEK OPERATION FAILED
4999                    BAD_NEC      EQU    020H         ; DISKETTE CONTROLLER HAS FAILED
5000                    BAD_CRC      EQU    010H         ; BAD CRC ON DISKETTE READ
5001                    MED_NOT_FND  EQU    00CH         ; MEDIA TYPE NOT FOUND
5002                    DMA_BOUNDARY EQU    009H         ; ATTEMPT TO DMA ACROSS 64K BOUNDARY
5003                    BAD_DMA      EQU    008H         ; DMA OVERRUN ON OPERATION
5004                    MEDIA_CHANGE EQU    006H         ; MEDIA REMOVED ON DUAL ATTACH CARD
5005                    RECORD_NOT_FND     EQU    004H         ; REQUESTED SECTOR NOT FOUND
5006                    WRITE_PROTECT      EQU    003H         ; WRITE ATTEMPTED ON WRITE PROTECT DISK
5007                    BAD_ADDR_MARK      EQU    002H         ; ADDRESS MARK NOT FOUND
5008                    BAD_CMD      EQU    001H         ; BAD COMMAND PASSED TO DISKETTE I/O
5009
5010                    ;---------- DISK CHANGE LINE EQUATES -------------------------------------------
5011                    NOCHGLN      EQU    001H         ; NO DISK CHANGE LINE AVAILABLE
5012                    CHGLN        EQU    002H         ; DISK CHANGE LINE AVAILABLE
5013
5014                    ;---------- MEDIA/DRIVE STATE INDICATORS ----------------------------------------
5015                    TRK_CAPA     EQU    00000001B    ; 80 TRACK CAPABILITY
5016                    FMT_CAPA     EQU    00000010B    ; MULTIPLE FORMAT CAPABILITY (1.2M)
5017                    DRV_DET      EQU    00000100B    ; DRIVE DETERMINED
5018                    MED_DET      EQU    00010000B    ; MEDIA DETERMINED BIT
```

```
                        DBL_STEP    EQU   00100000B    ; DOUBLE STEP BIT
                        RATE_MSK    EQU   11000000B    ; MASK FOR CLEARING ALL BUT RATE
                        RATE_500    EQU   00000000B    ; 500 KBS DATA RATE
                        RATE_300    EQU   01000000B    ; 300 KBS DATA RATE
                        RATE_250    EQU   10000000B    ; 250 KBS DATA RATE
                        STRT_MSK    EQU   00001100B    ; OPERATION START RATE MASK
                        SEND_MSK    EQU   11000000B    ; MASK FOR SEND RATE BITS

                        ;---------- MEDIA/DRIVE STATE INDICATORS COMPATIBILITY -------------------------
                        M3D3U       EQU   00000000B    ; 360 MEDIA/DRIVE NOT ESTABLISHED
                        M3D1U       EQU   00000001B    ; 360 MEDIA,1.2DRIVE NOT ESTABLISHED
                        M1D1U       EQU   00000010B    ; 1.2 MEDIA/DRIVE NOT ESTABLISHED
                        MED_UNK     EQU   00000111B    ; NONE OF THE ABOVE

                        ;---------- INTERRUPT EQUATES --------------------------------------------------
                        ;EOI        EQU   020H         ; END OF INTERRUPT COMMAND TO 8259
                        ;INTA00            EQU   020H         ; 8259 PORT
                        INTA01             EQU   021H         ; 8259 PORT
                        INTB00             EQU   0A0H         ; 2ND 8259
                        INTB01             EQU   0A1H         ;

                        ;-------------------------------------------------------------------------------
                        DMA08       EQU   008H         ; DMA STATUS REGISTER PORT ADDRESS
                        DMA         EQU   000H         ; DMA CH.0 ADDRESS REGISTER PORT ADDRESS
                        DMA18       EQU   0D0H         ; 2ND DMA STATUS PORT ADDRESS
                        DMA1        EQU   0C0H         ; 2ND DMA CH.0 ADDRESS REGISTER ADDRESS
                        ;-------------------------------------------------------------------------------
                        ;TIMER             EQU   040H         ; 8254 TIMER - BASE ADDRESS

                        ;-------------------------------------------------------------------------------
                        DMA_PAGE    EQU   081H         ; START OF DMA PAGE REGISTERS

                        ; 06/02/2015 (unix386.s, protected mode modifications)
                        ; (unix386.s <-- dsectrm2.s)
                        ; 11/12/2014 (copy from IBM PC-XT Model 286 BIOS - DSEG.INC)

                        ;---------------------------------------
                        ;     80286 INTERRUPT LOCATIONS :
                        ;     REFERENCED BY POST & BIOS :
                        ;---------------------------------------

00001DBE [3A1E0000]     DISK_POINTER:      dd     MD_TBL6            ; Pointer to Diskette Parameter Table

                        ; IBM PC-XT Model 286 source code ORGS.ASM (06/10/85) - 14/12/2014
                        ;-------------------------------------------------------------
                        ; DISK_BASE                                          :
                        ;     THIS IS THE SET OF PARAMETERS REQUIRED FOR          :
                        ;     DISKETTE OPERATION. THEY ARE POINTED AT BY THE            :
                        ;     DATA VARIABLE @DISK_POINTER. TO MODIFY THE PARAMETERS,     :
```

```
                                ;     BUILD ANOTHER PARAMETER BLOCK AND POINT AT IT       :
                                ;-----------------------------------------------------------

                                ;DISK_BASE:
                                ;     DB    11011111B    ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
                                ;     DB    2            ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
                                ;     DB    MOTOR_WAIT   ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
                                ;     DB    2            ; 512 BYTES/SECTOR
                                ;     ;DB   15           ; EOT (LAST SECTOR ON TRACK)
                                ;     db    18           ; (EOT for 1.44MB diskette)
                                ;     DB    01BH         ; GAP LENGTH
                                ;     DB    0FFH         ; DTL
                                ;     ;DB   054H         ; GAP LENGTH FOR FORMAT
                                ;     db    06ch         ; (for 1.44MB dsikette)
                                ;     DB    0F6H         ; FILL BYTE FOR FORMAT
                                ;     DB    15           ; HEAD SETTLE TIME (MILLISECONDS)
                                ;     DB    8            ; MOTOR START TIME (1/8 SECONDS)

                                ;---------------------------------------
                                ;     ROM BIOS DATA AREAS       :
                                ;---------------------------------------

                                ;DATA       SEGMENT AT 40H            ; ADDRESS= 0040:0000

                                ;@EQUIP_FLAG DW    ?            ; INSTALLED HARDWARE FLAGS

                                ;---------------------------------------
                                ;     DISKETTE DATA AREAS       :
                                ;---------------------------------------

                                ;@SEEK_STATUS      DB    ?           ; DRIVE RECALIBRATION STATUS
                                ;                                    ; BIT 3-0 = DRIVE 3-0 RECALIBRATION
                                ;                                    ; BEFORE NEXT SEEK IF BIT IS = 0
                                ;@MOTOR_STATUS     DB    ?           ; MOTOR STATUS
                                ;                                    ; BIT 3-0 = DRIVE 3-0 CURRENTLY RUNNING
                                ;                                    ; BIT 7 = CURRENT OPERATION IS A WRITE
                                ;@MOTOR_COUNT      DB    ?           ; TIME OUT COUNTER FOR MOTOR(S) TURN OFF
                                ;@DSKETTE_STATUS DB      ?           ; RETURN CODE STATUS BYTE
                                ;                                    ; CMD_BLOCK  IN STACK FOR DISK OPERATION
                                ;@NEC_STATUS DB    7 DUP(?)     ; STATUS BYTES FROM DISKETTE OPERATION

00001DC2 00                     SEEK_STATUS: db    0
00001DC3 00                     MOTOR_STATUS:      db    0
00001DC4 00                     MOTOR_COUNT: db    0
00001DC5 00                     DSKETTE_STATUS:    db    0
00001DC6 00000000000000         NEC_STATUS:  db    0,0,0,0,0,0,0

                                ;---------------------------------------
                                ;     POST AND BIOS WORK DATA AREA     :
```

```
                            ;----------------------------------------
                            ;@INTR_FLAG  DB    ?            ; FLAG INDICATING AN INTERRUPT HAPPENED
                            ;----------------------------------------
                            ;     TIMER DATA AREA           :
                            ;----------------------------------------
                            TIMER_LH:    ; 16/02/205
00001DCD 0000               TIMER_LOW:      DW      0             ; LOW WORD OF TIMER COUNT
00001DCF 0000               TIMER_HIGH:     DW      0             ; HIGH WORD OF TIMER COUNT
00001DD1 00                 TIMER_OFL:      DB      0             ; TIMER HAS ROLLED OVER SINCE LAST READ

                            ; 17/12/2014  (IRQ 0 - INT 08H)
                            ;TIMER_LOW  equ    46Ch          ; Timer ticks (counter)  @ 40h:006Ch
                            ;TIMER_HIGH equ    46Eh          ; (18.2 timer ticks per second)
                            ;TIMER_OFL  equ    470h          ; Timer - 24 hours flag  @ 40h:0070h

                            ;----------------------------------------
                            ;     ADDITIONAL MEDIA DATA           :
                            ;----------------------------------------

                            ;@LASTRATE   DB     ?            ; LAST DISKETTE DATA RATE SELECTED
                            ;@DSK_STATE  DB     ?            ; DRIVE 0 MEDIA STATE
                            ;            DB     ?            ; DRIVE 1 MEDIA STATE
                            ;            DB     ?            ; DRIVE 0 OPERATION START STATE
                            ;            DB     ?            ; DRIVE 1 OPERATION START STATE
                            ;@DSK_TRK    DB     ?            ; DRIVE 0 PRESENT CYLINDER
                            ;            DB     ?            ; DRIVE 1 PRESENT CYLINDER

00001DD2 00                 LASTRATE:    db     0
                            ;HF_STATUS:  db     0
                            ;HF_ERROR:   db     0
                            ;HF_INT_FLAG:       db     0
                            ;HF_CNTRL:   db     0
00001DD3 00000000           DSK_STATE:   db     0,0,0,0
00001DD7 0000               DSK_TRK:     db     0,0

                            ;DATA        ENDS               ; END OF BIOS DATA SEGMENT


                            ; 17/12/2014 (mov ax, [cfd])
                            ; 11/12/2014
00001DD9 00                 cfd:         db 0               ; current floppy drive (for GET_PARM)
                            ; 17/12/2014                    ; instead of 'DISK_POINTER'
00001DDA 01                 pfd:         db 1               ; previous floppy drive (for GET_PARM)
                                                            ; (initial value of 'pfd
                                                            ; must be different then 'cfd' value
                                                            ; to force updating/initializing
```

```
                                        ; current drive parameters)
; 10/12/2014
;
;int40h:
;     pushf
;     push  cs
;     ;cli
;     call  DISKETTE_IO_1
;     retn

; DSKETTE ----- 04/21/86 DISKETTE BIOS
; (IBM PC XT Model 286 System BIOS Source Code, 04-21-86)
;

;-- INT13H ----------------------------------------------------------------------
; DISKETTE I/O
;     THIS INTERFACE PROVIDES ACCESS TO THE 5 1/4 INCH 360 KB,
;     1.2 MB, 720 KB AND 1.44 MB DISKETTE DRIVES.
; INPUT
;     (AH) =  00H RESET DISKETTE SYSTEM
;           HARD RESET TO NEC, PREPARE COMMAND, RECALIBRATE REQUIRED
;           ON ALL DRIVES
;-------------------------------------------------------------------------------
;     (AH)= 01H  READ THE STATUS OF THE SYSTEM INTO (AH)
;           @DISKETTE_STATUS FROM LAST OPERATION IS USED
;-------------------------------------------------------------------------------
;     REGISTERS FOR READ/WRITE/VERIFY/FORMAT
;     (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
;     (DH) - HEAD NUMBER (0-1 ALLOWED, NOT VALUE CHECKED)
;     (CH) - TRACK NUMBER (NOT VALUE CHECKED)
;           MEDIA  DRIVE  TRACK NUMBER
;           320/360      320/360          0-39
;           320/360      1.2M       0-39
;           1.2M   1.2M       0-79
;           720K   720K       0-79
;           1.44M  1.44M      0-79
;     (CL) -       SECTOR NUMBER (NOT VALUE CHECKED, NOT USED FOR FORMAT)
;           MEDIA  DRIVE  SECTOR NUMBER
;           320/360      320/360           1-8/9
;           320/360      1.2M        1-8/9
;           1.2M   1.2M       1-15
;           720K   720K       1-9
;           1.44M  1.44M      1-18
;     (AL)  NUMBER OF SECTORS (NOT VALUE CHECKED)
;           MEDIA  DRIVE  MAX NUMBER OF SECTORS
;           320/360      320/360              8/9
;           320/360      1.2M          8/9
;           1.2M   1.2M         15
;           720K   720K         9
```

```
;           1.44M  1.44M       18
;
;     (ES:BX) - ADDRESS OF BUFFER (NOT REQUIRED FOR VERIFY)
;
;-------------------------------------------------------------------------------
;     (AH)= 02H  READ THE DESIRED SECTORS INTO MEMORY
;-------------------------------------------------------------------------------
;     (AH)= 03H  WRITE THE DESIRED SECTORS FROM MEMORY
;-------------------------------------------------------------------------------
;     (AH)= 04H  VERIFY THE DESIRED SECTORS
;-------------------------------------------------------------------------------
;     (AH)= 05H  FORMAT THE DESIRED TRACK
;           (ES,BX) MUST POINT TO THE COLLECTION OF DESIRED ADDRESS FIELDS
;           FOR THE      TRACK. EACH FIELD IS COMPOSED OF 4 BYTES, (C,H,R,N),
;           WHERE C = TRACK NUMBER, H=HEAD NUMBER, R = SECTOR NUMBER,
;           N= NUMBER OF BYTES PER SECTOR (00=128,01=256,02=512,03=1024),
;           THERE MUST BE ONE ENTRY FOR EVERY SECTOR ON THE TRACK.
;           THIS INFORMATION IS USED TO FIND THE REQUESTED SECTOR DURING
;           READ/WRITE ACCESS.
;           PRIOR TO FORMATTING A DISKETTE, IF THERE EXISTS MORE THAN
;           ONE SUPPORTED MEDIA FORMAT TYPE WITHIN THE DRIVE IN QUESTION,
;           THEN "SET DASD TYPE" (INT 13H, AH = 17H) OR 'SET MEDIA TYPE'
;           (INT 13H, AH =  18H) MUST BE CALLED TO SET THE DISKETTE TYPE
;           THAT IS TO BE FORMATTED. IF "SET DASD TYPE" OR "SET MEDIA TYPE"
;           IS NOT CALLED, THE FORMAT ROUTINE WILL ASSUME THE
;           MEDIA FORMAT TO BE THE MAXIMUM CAPACITY OF THE DRIVE.
;
;           THESE PARAMETERS OF DISK BASE MUST BE CHANGED IN ORDER TO
;           FORMAT THE FOLLOWING MEDIAS:
;           ------------------------------------------
;           : MEDIA  :     DRIVE      : PARM 1 : PARM 2 :
;           ------------------------------------------
;           : 320K  : 320K/360K/1.2M :  50H   :   8    :
;           : 360K  : 320K/360K/1.2M :  50H   :   9    :
;           : 1.2M  : 1.2M           :  54H   :  15    :
;           : 720K  : 720K/1.44M     :  50H   :   9    :
;           : 1.44M       : 1.44M          :  6CH   :  18    :
;           ------------------------------------------
;           NOTES: - PARM 1 = GAP LENGTH FOR FORMAT
;                  - PARM 2 = EOT (LAST SECTOR ON TRACK)
;                  - DISK BASE IS POINTED BY DISK POINTER LOCATED
;                   AT ABSOLUTE ADDRESS 0:78.
;                  - WHEN FORMAT OPERATIONS ARE COMPLETE, THE PARAMETERS
;                   SHOULD BE RESTORED TO THEIR RESPECTIVE INITIAL VALUES.

;-------------------------------------------------------------------------------
;     (AH) = 08H READ DRIVE PARAMETERS
;     REGISTERS
;       INPUT
```

```
;         (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
;       OUTPUT
;         (ES:DI) POINTS TO DRIVE PARAMETER TABLE
;         (CH) - LOW ORDER 8 OF 10 BITS MAXIMUM NUMBER OF TRACKS
;         (CL) - BITS 7 & 6 - HIGH ORDER TWO BITS OF MAXIMUM TRACKS
;                BITS 5 THRU 0 - MAXIMUM SECTORS PER TRACK
;         (DH) - MAXIMUM HEAD NUMBER
;         (DL) - NUMBER OF DISKETTE DRIVES INSTALLED
;         (BH) - 0
;         (BL) - BITS 7 THRU 4 - 0
;                BITS 3 THRU 0 - VALID DRIVE TYPE VALUE IN CMOS
;         (AX) - 0
;      UNDER THE FOLLOWING CIRCUMSTANCES:
;         (1) THE DRIVE NUMBER IS INVALID,
;         (2) THE DRIVE TYPE IS UNKNOWN AND CMOS IS NOT PRESENT,
;         (3) THE DRIVE TYPE IS UNKNOWN AND CMOS IS BAD,
;         (4) OR THE DRIVE TYPE IS UNKNOWN AND THE CMOS DRIVE TYPE IS INVALID
;         THEN ES,AX,BX,CX,DH,DI=0 ; DL=NUMBER OF DRIVES.
;         IF NO DRIVES ARE PRESENT THEN: ES,AX,BX,CX,DX,DI=0.
;         @DISKETTE_STATUS = 0 AND CY IS RESET.
;--------------------------------------------------------------------------------
;     (AH)= 15H  READ DASD TYPE
;     OUTPUT REGISTERS
;     (AH) - ON RETURN IF CARRY FLAG NOT SET, OTHERWISE ERROR
;            00 - DRIVE NOT PRESENT
;            01 - DISKETTE, NO CHANGE LINE AVAILABLE
;            02 - DISKETTE, CHANGE LINE AVAILABLE
;            03 - RESERVED (FIXED DISK)
;     (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
;--------------------------------------------------------------------------------
;     (AH)= 16H  DISK CHANGE LINE STATUS
;     OUTPUT REGISTERS
;     (AH) - 00 - DISK CHANGE LINE NOT ACTIVE
;            06 - DISK CHANGE LINE ACTIVE & CARRY BIT ON
;     (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED)
;--------------------------------------------------------------------------------
;     (AH)= 17H  SET DASD TYPE FOR FORMAT
;     INPUT REGISTERS
;     (AL) - 00 - NOT USED
;            01 - DISKETTE 320/360K IN 360K DRIVE
;            02 - DISKETTE 360K IN 1.2M DRIVE
;            03 - DISKETTE 1.2M IN 1.2M DRIVE
;            04 - DISKETTE 720K IN 720K DRIVE
;     (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHECKED:
;            (DO NOT USE WHEN DISKETTE ATTACH CARD USED)
;--------------------------------------------------------------------------------
;     (AH)= 18H  SET MEDIA TYPE FOR FORMAT
;     INPUT REGISTERS
;     (CH) - LOW ORDER 8 OF 10 BITS MAXIMUM TRACKS
```

```
;     (CL) - BITS 7 & 6 - HIGH ORDER TWO BITS OF MAXIMUM TRACKS
;            BITS 5 THRU 0 - MAXIMUM SECTORS PER TRACK
;     (DL) - DRIVE NUMBER (0-1 ALLOWED, VALUE CHACKED)
;     OUTPUT REGISTERS:
;     (ES:DI) - POINTER TO DRIVE PARAMETERS TABLE FOR THIS MEDIA TYPE,
;               UNCHANGED IF (AH) IS NON-ZERO
;     (AH) - 00H, CY = 0, TRACK AND SECTORS/TRACK COMBINATION IS SUPPORTED
;          - 01H, CY = 1, FUNCTION IS NOT AVAILABLE
;          - 0CH, CY = 1, TRACK AND SECTORS/TRACK COMBINATION IS NOT SUPPORTED
;          - 80H, CY = 1, TIME OUT (DISKETTE NOT PRESENT)
;--------------------------------------------------------------------------------
;     DISK CHANGE STATUS IS ONLY CHECKED WHEN A MEDIA SPECIFIED IS OTHER
;     THAN 360 KB DRIVE. IF THE DISK CHANGE LINE IS FOUND TO BE
;     ACTIVE THE FOLLOWING ACTIONS TAKE PLACE:
;           ATTEMPT TO RESET DISK CHANGE LINE TO INACTIVE STATE.
;           IF ATTEMPT SUCCEEDS SET DASD TYPE FOR FORMAT AND RETURN DISK
;           CHANGE ERROR CODE
;           IF ATTEMPT FAILS RETURN TIMEOUT ERROR CODE AND SET DASD TYPE
;           TO A PREDETERMINED STATE INDICATING MEDIA TYPE UNKNOWN.
;     IF THE DISK CHANGE LINE IN INACTIVE PERFORM SET DASD TYPE FOR FORMAT.
;
; DATA VARIABLE -- @DISK_POINTER
;     DOUBLE WORD POINTER TO THE CURRENT SET OF DISKETTE PARAMETERS
;--------------------------------------------------------------------------------
; OUTPUT FOR ALL FUNCTIONS
;     AH = STATUS OF OPERATION
;           STATUS BITS ARE DEFINED IN THE EQUATES FOR @DISKETTE_STATUS
;           VARIABLE IN THE DATA SEGMENT OF THIS MODULE
;     CY = 0 SUCCESSFUL OPERATION (AH=0 ON RETURN, EXCEPT FOR READ DASD
;           TYPE AH=(15)).
;     CY = 1 FAILED OPERATION (AH HAS ERROR REASON)
;     FOR READ/WRITE/VERIFY
;           DS,BX,DX,CX PRESERVED
;     NOTE: IF AN ERROR IS REPORTED BY THE DISKETTE CODE, THE APPROPRIATE
;           ACTION IS TO RESET THE DISKETTE, THEN RETRY THE OPERATION.
;           ON READ ACCESSES, NO MOTOR START DELAY IS TAKEN, SO THAT
;           THREE RETRIES ARE REQUIRED ON READS TO ENSURE THAT THE
;           PROBLEM IS NOT DUE TO MOTOR START-UP.
;--------------------------------------------------------------------------------
;
; DISKETTE STATE MACHINE - ABSOLUTE ADDRESS 40:90 (DRIVE A) & 91 (DRIVE B)
;
;   ---------------------------------------------------------------
;   |       |       |       |       |       |       |       |       |
;   |   7   |   6   |   5   |   4   |   3   |   2   |   1   |   0   |
;   |       |       |       |       |       |       |       |       |
;   ---------------------------------------------------------------
;     |     |     |     |     |     |     |     |
;     |     |     |     |     |     -----------------
```

```
                                ;     |      |      |      |      |           |
                                ;     |      |      |      |    RESERVED      |
                                ;     |      |      |      |              PRESENT STATE
                                ;     |      |      |      |      000: 360K IN 360K DRIVE UNESTABLISHED
                                ;     |      |      |      |      001: 360K IN 1.2M DRIVE UNESTABLISHED
                                ;     |      |      |      |      010: 1.2M IN 1.2M DRIVE UNESTABLISHED
                                ;     |      |      |      |      011: 360K IN 360K DRIVE ESTABLISHED
                                ;     |      |      |      |      100: 360K IN 1.2M DRIVE ESTABLISHED
                                ;     |      |      |      |      101: 1.2M IN 1.2M DRIVE ESTABLISHED
                                ;     |      |      |      |      110: RESERVED
                                ;     |      |      |      |      111: NONE OF THE ABOVE
                                ;     |      |      |      |
                                ;     |      |      |      ------>      MEDIA/DRIVE ESTABLISHED
                                ;     |      |      |
                                ;     |      |      --------------->    DOUBLE STEPPING REQUIRED (360K IN 1.2M
                                ;     |      |                          DRIVE)
                                ;     |      |
                                ;     ------------------------------>  DATA TRANSFER RATE FOR THIS DRIVE:
                                ;
                                ;                                       00: 500 KBS
                                ;                                       01: 300 KBS
                                ;                                       10: 250 KBS
                                ;                                       11: RESERVED
                                ;
                                ;
                                ;-------------------------------------------------------------------------
                                ; STATE OPERATION STARTED - ABSOLUTE ADDRESS 40:92 (DRIVE A) & 93 (DRIVE B)
                                ;-------------------------------------------------------------------------
                                ; PRESENT CYLINDER NUMBER - ABSOLUTE ADDRESS 40:94 (DRIVE A) & 95 (DRIVE B)
                                ;-------------------------------------------------------------------------

                                struc MD
00000000 <res 00000001>               .SPEC1       resb  1     ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
00000001 <res 00000001>               .SPEC2       resb  1     ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
00000002 <res 00000001>               .OFF_TIM     resb  1     ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
00000003 <res 00000001>               .BYT_SEC     resb  1     ; 512 BYTES/SECTOR
00000004 <res 00000001>               .SEC_TRK     resb  1     ; EOT (LAST SECTOR ON TRACK)
00000005 <res 00000001>               .GAP         resb  1     ; GAP LENGTH
00000006 <res 00000001>               .DTL         resb  1     ; DTL
00000007 <res 00000001>               .GAP3        resb  1     ; GAP LENGTH FOR FORMAT
00000008 <res 00000001>               .FIL_BYT     resb  1     ; FILL BYTE FOR FORMAT
00000009 <res 00000001>               .HD_TIM            resb  1     ; HEAD SETTLE TIME (MILLISECONDS)
0000000A <res 00000001>               .STR_TIM     resb  1     ; MOTOR START TIME (1/8 SECONDS)
0000000B <res 00000001>               .MAX_TRK     resb  1     ; MAX. TRACK NUMBER
0000000C <res 00000001>               .RATE        resb  1     ; DATA TRANSFER RATE
                                endstruc

                                BIT7OFF     EQU   7FH
                                BIT7ON      EQU   80H
```

```
5410
5411                                    ;--------------------------------------------------
5412                                    ;     DRIVE TYPE TABLE                      :
5413                                    ;--------------------------------------------------
5414                                            ; 16/02/2015 (unix386.s, 32 bit modifications)
5415                                    DR_TYPE:
5416 00001DDB 01                               DB     01           ;DRIVE TYPE, MEDIA TABLE
5417                                               ;DW      MD_TBL1
5418 00001DDC [F91D0000]                       dd     MD_TBL1
5419 00001DE0 82                               DB     02+BIT7ON
5420                                            ;DW      MD_TBL2
5421 00001DE1 [061E0000]                          dd      MD_TBL2
5422 00001DE5 02                       DR_DEFAULT:  DB     02
5423                                               ;DW      MD_TBL3
5424 00001DE6 [131E0000]                       dd      MD_TBL3
5425 00001DEA 03                               DB     03
5426                                               ;DW      MD_TBL4
5427 00001DEB [201E0000]                       dd      MD_TBL4
5428 00001DEF 84                               DB     04+BIT7ON
5429                                               ;DW      MD_TBL5
5430 00001DF0 [2D1E0000]                       dd      MD_TBL5
5431 00001DF4 04                               DB     04
5432                                               ;DW      MD_TBL6
5433 00001DF5 [3A1E0000]                       dd      MD_TBL6
5434                                    DR_TYPE_E       equ $                   ; END OF TABLE
5435                                    ;DR_CNT            EQU    (DR_TYPE_E-DR_TYPE)/3
5436                                    DR_CNT             equ    (DR_TYPE_E-DR_TYPE)/5
5437                                    ;--------------------------------------------------
5438                                    ;     MEDIA/DRIVE PARAMETER TABLES               :
5439                                    ;--------------------------------------------------
5440                                    ;--------------------------------------------------
5441                                    ;     360 KB MEDIA IN 360 KB DRIVE               :
5442                                    ;--------------------------------------------------
5443                                    MD_TBL1:
5444 00001DF9 DF                               DB     11011111B    ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
5445 00001DFA 02                               DB     2            ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
5446 00001DFB 25                               DB     MOTOR_WAIT   ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
5447 00001DFC 02                               DB     2            ; 512 BYTES/SECTOR
5448 00001DFD 09                               DB     09           ; EOT (LAST SECTOR ON TRACK)
5449 00001DFE 2A                               DB     02AH         ; GAP LENGTH
5450 00001DFF FF                               DB     0FFH         ; DTL
5451 00001E00 50                               DB     050H         ; GAP LENGTH FOR FORMAT
5452 00001E01 F6                               DB     0F6H         ; FILL BYTE FOR FORMAT
5453 00001E02 0F                               DB     15           ; HEAD SETTLE TIME (MILLISECONDS)
5454 00001E03 08                               DB     8            ; MOTOR START TIME (1/8 SECONDS)
5455 00001E04 27                               DB     39           ; MAX. TRACK NUMBER
5456 00001E05 80                               DB     RATE_250     ; DATA TRANSFER RATE
5457                                    ;--------------------------------------------------
5458                                    ;     360 KB MEDIA IN 1.2 MB DRIVE               :
```

```
5459                                    ;--------------------------------------------------------
5460                                    MD_TBL2:
5461 00001E06 DF                              DB    11011111B    ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
5462 00001E07 02                              DB    2            ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
5463 00001E08 25                              DB    MOTOR_WAIT   ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
5464 00001E09 02                              DB    2            ; 512 BYTES/SECTOR
5465 00001E0A 09                              DB    09           ; EOT (LAST SECTOR ON TRACK)
5466 00001E0B 2A                              DB    02AH         ; GAP LENGTH
5467 00001E0C FF                              DB    0FFH         ; DTL
5468 00001E0D 50                              DB    050H         ; GAP LENGTH FOR FORMAT
5469 00001E0E F6                              DB    0F6H         ; FILL BYTE FOR FORMAT
5470 00001E0F 0F                              DB    15           ; HEAD SETTLE TIME (MILLISECONDS)
5471 00001E10 08                              DB    8            ; MOTOR START TIME (1/8 SECONDS)
5472 00001E11 27                              DB    39           ; MAX. TRACK NUMBER
5473 00001E12 40                              DB    RATE_300     ; DATA TRANSFER RATE
5474                                    ;--------------------------------------------------------
5475                                    ;     1.2 MB MEDIA IN 1.2 MB DRIVE                  :
5476                                    ;--------------------------------------------------------
5477                                    MD_TBL3:
5478 00001E13 DF                              DB    11011111B    ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
5479 00001E14 02                              DB    2            ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
5480 00001E15 25                              DB    MOTOR_WAIT   ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
5481 00001E16 02                              DB    2            ; 512 BYTES/SECTOR
5482 00001E17 0F                              DB    15           ; EOT (LAST SECTOR ON TRACK)
5483 00001E18 1B                              DB    01BH         ; GAP LENGTH
5484 00001E19 FF                              DB    0FFH         ; DTL
5485 00001E1A 54                              DB    054H         ; GAP LENGTH FOR FORMAT
5486 00001E1B F6                              DB    0F6H         ; FILL BYTE FOR FORMAT
5487 00001E1C 0F                              DB    15           ; HEAD SETTLE TIME (MILLISECONDS)
5488 00001E1D 08                              DB    8            ; MOTOR START TIME (1/8 SECONDS)
5489 00001E1E 4F                              DB    79           ; MAX. TRACK NUMBER
5490 00001E1F 00                              DB    RATE_500     ; DATA TRANSFER RATE
5491                                    ;--------------------------------------------------------
5492                                    ;     720 KB MEDIA IN 720 KB DRIVE                  :
5493                                    ;--------------------------------------------------------
5494                                    MD_TBL4:
5495 00001E20 DF                              DB    11011111B    ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
5496 00001E21 02                              DB    2            ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
5497 00001E22 25                              DB    MOTOR_WAIT   ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
5498 00001E23 02                              DB    2            ; 512 BYTES/SECTOR
5499 00001E24 09                              DB    09           ; EOT (LAST SECTOR ON TRACK)
5500 00001E25 2A                              DB    02AH         ; GAP LENGTH
5501 00001E26 FF                              DB    0FFH         ; DTL
5502 00001E27 50                              DB    050H         ; GAP LENGTH FOR FORMAT
5503 00001E28 F6                              DB    0F6H         ; FILL BYTE FOR FORMAT
5504 00001E29 0F                              DB    15           ; HEAD SETTLE TIME (MILLISECONDS)
5505 00001E2A 08                              DB    8            ; MOTOR START TIME (1/8 SECONDS)
5506 00001E2B 4F                              DB    79           ; MAX. TRACK NUMBER
5507 00001E2C 80                              DB    RATE_250     ; DATA TRANSFER RATE
```

```
5508                                    ;--------------------------------------------------------
5509                                    ;     720 KB MEDIA IN 1.44 MB DRIVE                :
5510                                    ;--------------------------------------------------------
5511                                    MD_TBL5:
5512 00001E2D DF                               DB    11011111B    ; SRT=D, HD UNLOAD=0F - 1ST SPECIFY BYTE
5513 00001E2E 02                               DB    2            ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
5514 00001E2F 25                               DB    MOTOR_WAIT   ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
5515 00001E30 02                               DB    2            ; 512 BYTES/SECTOR
5516 00001E31 09                               DB    09           ; EOT (LAST SECTOR ON TRACK)
5517 00001E32 2A                               DB    02AH         ; GAP LENGTH
5518 00001E33 FF                               DB    0FFH         ; DTL
5519 00001E34 50                               DB    050H         ; GAP LENGTH FOR FORMAT
5520 00001E35 F6                               DB    0F6H         ; FILL BYTE FOR FORMAT
5521 00001E36 0F                               DB    15           ; HEAD SETTLE TIME (MILLISECONDS)
5522 00001E37 08                               DB    8            ; MOTOR START TIME (1/8 SECONDS)
5523 00001E38 4F                               DB    79           ; MAX. TRACK NUMBER
5524 00001E39 80                               DB    RATE_250     ; DATA TRANSFER RATE
5525                                    ;--------------------------------------------------------
5526                                    ;     1.44 MB MEDIA IN 1.44 MB DRIVE               :
5527                                    ;--------------------------------------------------------
5528                                    MD_TBL6:
5529 00001E3A AF                               DB    10101111B    ; SRT=A, HD UNLOAD=0F - 1ST SPECIFY BYTE
5530 00001E3B 02                               DB    2            ; HD LOAD=1, MODE=DMA - 2ND SPECIFY BYTE
5531 00001E3C 25                               DB    MOTOR_WAIT   ; WAIT TIME AFTER OPERATION TILL MOTOR OFF
5532 00001E3D 02                               DB    2            ; 512 BYTES/SECTOR
5533 00001E3E 12                               DB    18           ; EOT (LAST SECTOR ON TRACK)
5534 00001E3F 1B                               DB    01BH         ; GAP LENGTH
5535 00001E40 FF                               DB    0FFH         ; DTL
5536 00001E41 6C                               DB    06CH         ; GAP LENGTH FOR FORMAT
5537 00001E42 F6                               DB    0F6H         ; FILL BYTE FOR FORMAT
5538 00001E43 0F                               DB    15           ; HEAD SETTLE TIME (MILLISECONDS)
5539 00001E44 08                               DB    8            ; MOTOR START TIME (1/8 SECONDS)
5540 00001E45 4F                               DB    79           ; MAX. TRACK NUMBER
5541 00001E46 00                               DB    RATE_500     ; DATA TRANSFER RATE
5542
5543                                    ;;int13h: ; 16/02/2015
5544                                    ;; 16/02/2015 - 21/02/2015
5545                                    int40h:
5546 00001E47 9C                               pushfd
5547 00001E48 0E                               push  cs
5548 00001E49 E801000000                       call  DISKETTE_IO_1
5549 00001E4E C3                               retn
5550
5551                                    DISKETTE_IO_1:
5552
5553 00001E4F FB                               STI                   ; INTERRUPTS BACK ON
5554 00001E50 55                               PUSH  eBP             ; USER REGISTER
5555 00001E51 57                               PUSH  eDI             ; USER REGISTER
5556 00001E52 52                               PUSH  eDX             ; HEAD #, DRIVE # OR USER REGISTER
```

```
5557 00001E53 53                        PUSH   eBX                ; BUFFER OFFSET PARAMETER OR REGISTER
5558 00001E54 51                        PUSH   eCX                ; TRACK #-SECTOR # OR USER REGISTER
5559 00001E55 89E5                      MOV    eBP,eSP                  ; BP     => PARAMETER LIST DEP. ON AH
5560                                                                ; [BP]   = SECTOR #
5561                                                                ; [BP+1] = TRACK #
5562                                                                ; [BP+2] = BUFFER OFFSET
5563                                                                ; FOR RETURN OF DRIVE PARAMETERS:
5564                                                                ; CL/[BP] = BITS 7&6 HI BITS OF MAX CYL
5565                                                                ;           BITS 0-5 MAX SECTORS/TRACK
5566                                                                ; CH/[BP+1] = LOW 8 BITS OF MAX CYL.
5567                                                                ; BL/[BP+2] = BITS 7-4 = 0
5568                                                                ;             BITS 3-0 = VALID CMOS TYPE
5569                                                                ; BH/[BP+3] = 0
5570                                                                ; DL/[BP+4] = # DRIVES INSTALLED
5571                                                                ; DH/[BP+5] = MAX HEAD #
5572                                                                ; DI/[BP+6] = OFFSET TO DISK BASE
5573 00001E57 06                        push   es ; 06/02/2015
5574 00001E58 1E                        PUSH   DS                 ; BUFFER SEGMENT PARM OR USER REGISTER
5575 00001E59 56                        PUSH   eSI                ; USER REGISTERS
5576                                    ;CALL  DDS                ; SEGMENT OF BIOS DATA AREA TO DS
5577                                    ;mov   cx, cs
5578                                    ;mov   ds, cx
5579 00001E5A 66B91000                  mov    cx, KDATA
5580 00001E5E 8ED9                        mov     ds, cx
5581 00001E60 8EC1                        mov     es, cx
5582
5583                                    ;CMP   AH,(FNC_TAE-FNC_TAB)/2    ; CHECK FOR > LARGEST FUNCTION
5584 00001E62 80FC19                    cmp    ah,(FNC_TAE-FNC_TAB)/4 ; 18/02/2015
5585 00001E65 7202                      JB     short OK_FUNC      ; FUNCTION OK
5586 00001E67 B414                      MOV    AH,14H             ; REPLACE WITH KNOWN INVALID FUNCTION
5587                                OK_FUNC:
5588 00001E69 80FC01                    CMP    AH,1               ; RESET OR STATUS ?
5589 00001E6C 760C                      JBE    short OK_DRV       ; IF RESET OR STATUS DRIVE ALWAYS OK
5590 00001E6E 80FC08                    CMP    AH,8               ; READ DRIVE PARMS ?
5591 00001E71 7407                      JZ     short OK_DRV       ; IF SO DRIVE CHECKED LATER
5592 00001E73 80FA01                    CMP    DL,1               ; DRIVES 0 AND 1 OK
5593 00001E76 7602                      JBE    short OK_DRV       ; IF 0 OR 1 THEN JUMP
5594 00001E78 B414                      MOV    AH,14H             ; REPLACE WITH KNOWN INVALID FUNCTION
5595                                OK_DRV:
5596 00001E7A 31C9                      xor    ecx, ecx
5597                                    ;mov   esi, ecx ; 08/02/2015
5598 00001E7C 89CF                      mov    edi, ecx ; 08/02/2015
5599 00001E7E 88E1                      MOV    CL,AH              ; CL = FUNCTION
5600                                    ;XOR   CH,CH              ; CX = FUNCTION
5601                                    ;SHL   CL, 1              ; FUNCTION TIMES 2
5602 00001E80 C0E102                    SHL    CL, 2 ; 20/02/2015  ; FUNCTION TIMES 4 (for 32 bit offset)
5603 00001E83 BB[BB1E0000]              MOV    eBX,FNC_TAB        ; LOAD START OF FUNCTION TABLE
5604 00001E88 01CB                      ADD    eBX,eCX                  ; ADD OFFSET INTO TABLE => ROUTINE
5605 00001E8A 88F4                      MOV    AH,DH              ; AX = HEAD #,# OF SECTORS OR DASD TYPE
```

```
5606 00001E8C 30F6                      XOR    DH,DH              ; DX = DRIVE #
5607 00001E8E 6689C6                    MOV    SI,AX              ; SI = HEAD #,# OF SECTORS OR DASD TYPE
5608 00001E91 6689D7                    MOV     DI,DX                   ; DI = DRIVE #
5609                                    ;
5610                                    ; 11/12/2014
5611 00001E94 8815[D91D0000]              mov     [cfd], dl               ; current floppy drive (for 'GET_PARM')
5612                                    ;
5613 00001E9A 8A25[C51D0000]            MOV    AH, [DSKETTE_STATUS]      ; LOAD STATUS TO AH FOR STATUS FUNCTION
5614 00001EA0 C605[C51D0000]00          MOV    byte [DSKETTE_STATUS],0   ; INITIALIZE FOR ALL OTHERS
5615
5616                                ;     THROUGHOUT THE DISKETTE BIOS, THE FOLLOWING INFORMATION IS CONTAINED IN
5617                                ;     THE FOLLOWING MEMORY LOCATIONS AND REGISTERS. NOT ALL DISKETTE BIOS
5618                                ;     FUNCTIONS REQUIRE ALL OF THESE PARAMETERS.
5619                                ;
5620                                ;           DI     : DRIVE #
5621                                ;           SI-HI  : HEAD #
5622                                ;           SI-LOW : # OF SECTORS OR DASD TYPE FOR FORMAT
5623                                ;           ES     : BUFFER SEGMENT
5624                                ;           [BP]   : SECTOR #
5625                                ;           [BP+1] : TRACK #
5626                                ;           [BP+2] : BUFFER OFFSET
5627                                ;
5628                                ;     ACROSS CALLS TO SUBROUTINES THE CARRY FLAG (CY=1), WHERE INDICATED IN
5629                                ;     SUBROUTINE PROLOGUES, REPRESENTS AN EXCEPTION RETURN (NORMALLY AN ERROR
5630                                ;     CONDITION). IN MOST CASES, WHEN CY = 1, @DSKETTE_STATUS CONTAINS THE
5631                                ;     SPECIFIC ERROR CODE.
5632                                ;
5633                                                               ; (AH) = @DSKETTE_STATUS
5634 00001EA7 FF13                      CALL   dWORD [eBX]         ; CALL THE REQUESTED FUNCTION
5635 00001EA9 5E                        POP    eSI                 ; RESTORE ALL REGISTERS
5636 00001EAA 1F                        POP    DS
5637 00001EAB 07                        pop    es     ; 06/02/2015
5638 00001EAC 59                        POP    eCX
5639 00001EAD 5B                        POP    eBX
5640 00001EAE 5A                        POP    eDX
5641 00001EAF 5F                        POP    eDI
5642 00001EB0 89E5                      MOV    eBP, eSP
5643 00001EB2 50                        PUSH   eAX
5644 00001EB3 9C                        PUSHFd
5645 00001EB4 58                        POP    eAX
5646                                    ;MOV   [BP+6], AX
5647 00001EB5 89450C                    mov    [ebp+12], eax  ; 18/02/2015, flags
5648 00001EB8 58                        POP    eAX
5649 00001EB9 5D                        POP    eBP
5650 00001EBA CF                        IRETd
5651
5652                                ;--------------------------------------------------------------------------
5653                                ; DW --> dd (06/02/2015)
5654 00001EBB [1F1F0000]            FNC_TAB     dd     DSK_RESET           ; AH = 00H; RESET
```

```
5655 00001EBF [8F1F0000]                dd     DSK_STATUS          ; AH = 01H; STATUS
5656 00001EC3 [A01F0000]                dd     DSK_READ            ; AH = 02H; READ
5657 00001EC7 [B11F0000]                dd     DSK_WRITE           ; AH = 03H; WRITE
5658 00001ECB [C21F0000]                dd     DSK_VERF            ; AH = 04H; VERIFY
5659 00001ECF [D31F0000]                dd     DSK_FORMAT          ; AH = 05H; FORMAT
5660 00001ED3 [58200000]                dd     FNC_ERR                   ; AH = 06H; INVALID
5661 00001ED7 [58200000]                dd     FNC_ERR                   ; AH = 07H; INVALID
5662 00001EDB [65200000]                dd     DSK_PARMS           ; AH = 08H; READ DRIVE PARAMETERS
5663 00001EDF [58200000]                dd     FNC_ERR                   ; AH = 09H; INVALID
5664 00001EE3 [58200000]                dd     FNC_ERR                   ; AH = 0AH; INVALID
5665 00001EE7 [58200000]                dd     FNC_ERR                   ; AH = 0BH; INVALID
5666 00001EEB [58200000]                dd     FNC_ERR                   ; AH = 0CH; INVALID
5667 00001EEF [58200000]                dd     FNC_ERR                   ; AH = 0DH; INVALID
5668 00001EF3 [58200000]                dd     FNC_ERR                   ; AH = 0EH; INVALID
5669 00001EF7 [58200000]                dd     FNC_ERR                   ; AH = 0FH; INVALID
5670 00001EFB [58200000]                dd     FNC_ERR                   ; AH = 10H; INVALID
5671 00001EFF [58200000]                dd     FNC_ERR                   ; AH = 11H; INVALID
5672 00001F03 [58200000]                dd     FNC_ERR                   ; AH = 12H; INVALID
5673 00001F07 [58200000]                dd     FNC_ERR                   ; AH = 13H; INVALID
5674 00001F0B [58200000]                dd     FNC_ERR                   ; AH = 14H; INVALID
5675 00001F0F [26210000]                dd     DSK_TYPE            ; AH = 15H; READ DASD TYPE
5676 00001F13 [51210000]                dd     DSK_CHANGE          ; AH = 16H; CHANGE STATUS
5677 00001F17 [8B210000]                dd     FORMAT_SET          ; AH = 17H; SET DASD TYPE
5678 00001F1B [0E220000]                dd     SET_MEDIA           ; AH = 18H; SET MEDIA TYPE
5679                                FNC_TAE EQU     $                    ; END
5680
5681                                ;--------------------------------------------------------------------------
5682                                ; DISK_RESET (AH = 00H)
5683                                ;           RESET THE DISKETTE SYSTEM.
5684                                ;
5685                                ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
5686                                ;--------------------------------------------------------------------------
5687                                DSK_RESET:
5688 00001F1F 66BAF203                  MOV    DX,03F2H             ; ADAPTER CONTROL PORT
5689 00001F23 FA                        CLI                         ; NO INTERRUPTS
5690 00001F24 A0[C31D0000]              MOV    AL,[MOTOR_STATUS]    ; GET DIGITAL OUTPUT REGISTER REFLECTION
5691 00001F29 243F                      AND    AL,00111111B         ; KEEP SELECTED AND MOTOR ON BITS
5692 00001F2B C0C004                    ROL    AL,4                 ; MOTOR VALUE TO HIGH NIBBLE
5693                                                                ; DRIVE SELECT TO LOW NIBBLE
5694 00001F2E 0C08                      OR     AL,00001000B         ; TURN ON INTERRUPT ENABLE
5695 00001F30 EE                        OUT    DX,AL                ; RESET THE ADAPTER
5696 00001F31 C605[C21D0000]00          MOV    byte [SEEK_STATUS],0      ; SET RECALIBRATE REQUIRED ON ALL DRIVES
5697                                    ;JMP   $+2                  ; WAIT FOR I/O
5698                                    ;JMP   $+2                  ; WAIT FOR I/O (TO INSURE MINIMUM
5699                                                                ;      PULSE WIDTH)
5700                                    ; 19/12/2014
5701                                    NEWIODELAY
5702 00001F38 E6EB                  <1>  out 0ebh,al
5703
```

```
                                        ; 17/12/2014
                                        ; AWARD BIOS 1999 - RESETDRIVES (ADISK.ASM)
                                ;       mov   cx, WAITCPU_RESET_ON      ; cx = 21 -- Min. 14 micro seconds !?
                                ;wdw1:
                                ;       NEWIODELAY
                                ;       loop  wdw1
                                        ;
00001F3A 0C04                           OR    AL,00000100B              ; TURN OFF RESET BIT
00001F3C EE                             OUT   DX,AL                     ; RESET THE ADAPTER
                                        ; 16/12/2014
                                        IODELAY
00001F3D EB00                   <1>  jmp short $+2
00001F3F EB00                   <1>  jmp short $+2
                                        ;
                                        ;STI                            ; ENABLE THE INTERRUPTS
00001F41 E81B0C0000                     CALL  WAIT_INT                  ; WAIT FOR THE INTERRUPT
00001F46 723E                           JC    short DR_ERR              ; IF ERROR, RETURN IT
00001F48 66B9C000                       MOV   CX,11000000B              ; CL = EXPECTED @NEC_STATUS
                                NXT_DRV:
00001F4C 6651                           PUSH  CX                        ; SAVE FOR CALL
00001F4E B8[841F0000]                   MOV   eAX, DR_POP_ERR           ; LOAD NEC_OUTPUT ERROR ADDRESS
00001F53 50                             PUSH  eAX                       ; "
00001F54 B408                           MOV   AH,08H                    ; SENSE INTERRUPT STATUS COMMAND
00001F56 E8040B0000                     CALL  NEC_OUTPUT
00001F5B 58                             POP   eAX                       ; THROW AWAY ERROR RETURN
00001F5C E82D0C0000                     CALL  RESULTS                        ; READ IN THE RESULTS
00001F61 6659                           POP   CX                        ; RESTORE AFTER CALL
00001F63 7221                           JC    short DR_ERR              ; ERROR RETURN
00001F65 3A0D[C61D0000]                 CMP   CL, [NEC_STATUS]          ; TEST FOR DRIVE READY TRANSITION
00001F6B 7519                           JNZ   short DR_ERR              ; EVERYTHING OK
00001F6D FEC1                           INC   CL                        ; NEXT EXPECTED @NEC_STATUS
00001F6F 80F9C3                         CMP   CL,11000011B              ; ALL POSSIBLE DRIVES CLEARED
00001F72 76D8                           JBE   short NXT_DRV             ; FALL THRU IF 11000100B OR >
                                        ;
00001F74 E852030000                     CALL  SEND_SPEC                 ; SEND SPECIFY COMMAND TO NEC
                                RESBAC:
00001F79 E807090000                     CALL  SETUP_END                 ; VARIOUS CLEANUPS
00001F7E 6689F3                         MOV   BX,SI                     ; GET SAVED AL TO BL
00001F81 88D8                           MOV   AL,BL                     ; PUT BACK FOR RETURN
00001F83 C3                             RETn
                                DR_POP_ERR:
00001F84 6659                           POP   CX                        ; CLEAR STACK
                                DR_ERR:
00001F86 800D[C51D0000]20               OR    byte [DSKETTE_STATUS],BAD_NEC ; SET ERROR CODE
00001F8D EBEA                           JMP   SHORT RESBAC              ; RETURN FROM RESET

                                ;--------------------------------------------------------------------------------
                                ; DISK_STATUS      (AH = 01H)
                                ;     DISKETTE STATUS.
```

```
                                ;
                                ; ON ENTRY:  AH : STATUS OF PREVIOUS OPERATION
                                ;
                                ; ON EXIT:   AH, @DSKETTE_STATUS, CY REFLECT STATUS OF PREVIOUS OPERATION.
                                ;--------------------------------------------------------------------------------
                                DSK_STATUS:
00001F8F 8825[C51D0000]               MOV    [DSKETTE_STATUS],AH ; PUT BACK FOR SETUP END
00001F95 E8EB080000                   CALL   SETUP_END           ; VARIOUS CLEANUPS
00001F9A 6689F3                       MOV    BX,SI               ; GET SAVED AL TO BL
00001F9D 88D8                         MOV    AL,BL               ; PUT BACK FOR RETURN
00001F9F C3                           RETn

                                ;--------------------------------------------------------------------------------
                                ; DISK_READ  (AH = 02H)
                                ;     DISKETTE READ.
                                ;
                                ; ON ENTRY:  DI     : DRIVE #
                                ;            SI-HI  : HEAD #
                                ;            SI-LOW : # OF SECTORS
                                ;            ES     : BUFFER SEGMENT
                                ;            [BP]   : SECTOR #
                                ;            [BP+1] : TRACK #
                                ;            [BP+2] : BUFFER OFFSET
                                ;
                                ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
                                ;--------------------------------------------------------------------------------

                                ; 06/02/2015, ES:BX -> EBX (unix386.s)

                                DSK_READ:
00001FA0 8025[C31D0000]7F             AND    byte [MOTOR_STATUS],01111111B ; INDICATE A READ OPERATION
00001FA7 66B846E6                     MOV    AX,0E646H           ; AX = NEC COMMAND, DMA COMMAND
00001FAB E825040000                   CALL   RD_WR_VF            ; COMMON READ/WRITE/VERIFY
00001FB0 C3                           RETn

                                ;--------------------------------------------------------------------------------
                                ; DISK_WRITE(AH = 03H)
                                ;     DISKETTE WRITE.
                                ;
                                ; ON ENTRY:  DI     : DRIVE #
                                ;            SI-HI  : HEAD #
                                ;            SI-LOW : # OF SECTORS
                                ;            ES     : BUFFER SEGMENT
                                ;            [BP]   : SECTOR #
                                ;            [BP+1] : TRACK #
                                ;            [BP+2] : BUFFER OFFSET
                                ;
                                ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
                                ;--------------------------------------------------------------------------------
```

```
5802
5803                                    ; 06/02/2015, ES:BX -> EBX (unix386.s)
5804
5805                                    DSK_WRITE:
5806 00001FB1 66B84AC5                       MOV   AX,0C54AH           ; AX = NEC COMMAND, DMA COMMAND
5807 00001FB5 800D[C31D0000]80                OR      byte [MOTOR_STATUS],10000000B ; INDICATE WRITE OPERATION
5808 00001FBC E814040000                     CALL  RD_WR_VF            ; COMMON READ/WRITE/VERIFY
5809 00001FC1 C3                             RETn
5810
5811                                    ;--------------------------------------------------------------------------
5812                                    ; DISK_VERF  (AH = 04H)
5813                                    ;     DISKETTE VERIFY.
5814                                    ;
5815                                    ; ON ENTRY:  DI     : DRIVE #
5816                                    ;            SI-HI  : HEAD #
5817                                    ;            SI-LOW : # OF SECTORS
5818                                    ;            ES     : BUFFER SEGMENT
5819                                    ;            [BP]   : SECTOR #
5820                                    ;            [BP+1] : TRACK #
5821                                    ;            [BP+2] : BUFFER OFFSET
5822                                    ;
5823                                    ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
5824                                    ;--------------------------------------------------------------------------
5825                                    DSK_VERF:
5826 00001FC2 8025[C31D0000]7F               AND   byte [MOTOR_STATUS],01111111B ; INDICATE A READ OPERATION
5827 00001FC9 66B842E6                       MOV   AX,0E642H           ; AX = NEC COMMAND, DMA COMMAND
5828 00001FCD E803040000                     CALL  RD_WR_VF            ; COMMON READ/WRITE/VERIFY
5829 00001FD2 C3                             RETn
5830
5831                                    ;--------------------------------------------------------------------------
5832                                    ; DISK_FORMAT      (AH = 05H)
5833                                    ;     DISKETTE FORMAT.
5834                                    ;
5835                                    ; ON ENTRY:  DI     : DRIVE #
5836                                    ;            SI-HI  : HEAD #
5837                                    ;            SI-LOW : # OF SECTORS
5838                                    ;            ES     : BUFFER SEGMENT
5839                                    ;            [BP]   : SECTOR #
5840                                    ;            [BP+1] : TRACK #
5841                                    ;            [BP+2] : BUFFER OFFSET
5842                                    ;            @DISK_POINTER POINTS TO THE PARAMETER TABLE OF THIS DRIVE
5843                                    ;
5844                                    ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
5845                                    ;--------------------------------------------------------------------------
5846                                    DSK_FORMAT:
5847 00001FD3 E83C030000                     CALL  XLAT_NEW            ; TRANSLATE STATE TO PRESENT ARCH.
5848 00001FD8 E838050000                     CALL  FMT_INIT            ; ESTABLISH STATE IF UNESTABLISHED
5849 00001FDD 800D[C31D0000]80                OR      byte [MOTOR_STATUS], 10000000B ; INDICATE WRITE OPERATION
5850 00001FE4 E880050000                     CALL  MED_CHANGE          ; CHECK MEDIA CHANGE AND RESET IF SO
```

```
5851 00001FE9 725D                      JC      short FM_DON            ; MEDIA CHANGED, SKIP
5852 00001FEB E8DB020000              CALL  SEND_SPEC           ; SEND SPECIFY COMMAND TO NEC
5853 00001FF0 E8E6050000              CALL  CHK_LASTRATE        ; ZF=1 ATTEMPT RATE IS SAME AS LAST RATE
5854 00001FF5 7405                      JZ      short FM_WR             ; YES, SKIP SPECIFY COMMAND
5855 00001FF7 E8BD050000              CALL  SEND_RATE           ; SEND DATA RATE TO CONTROLLER
5856                                  FM_WR:
5857 00001FFC E874060000              CALL  FMTDMA_SET          ; SET UP THE DMA FOR FORMAT
5858 00002001 7245                      JC      short FM_DON            ; RETURN WITH ERROR
5859 00002003 B44D                    MOV   AH,04DH                 ; ESTABLISH THE FORMAT COMMAND
5860 00002005 E8D1060000              CALL  NEC_INIT            ; INITIALIZE THE NEC
5861 0000200A 723C                      JC      short FM_DON            ; ERROR - EXIT
5862 0000200C B8[48200000]              MOV     eAX, FM_DON             ; LOAD ERROR ADDRESS
5863 00002011 50                      PUSH  eAX                 ; PUSH NEC_OUT ERROR RETURN
5864 00002012 B203                    MOV   DL,3                ; BYTES/SECTOR VALUE TO NEC
5865 00002014 E840090000              CALL  GET_PARM
5866 00002019 E8410A0000              CALL  NEC_OUTPUT
5867 0000201E B204                    MOV   DL,4                ; SECTORS/TRACK VALUE TO NEC
5868 00002020 E834090000              CALL  GET_PARM
5869 00002025 E8350A0000              CALL  NEC_OUTPUT
5870 0000202A B207                    MOV   DL,7                ; GAP LENGTH VALUE TO NEC
5871 0000202C E828090000              CALL  GET_PARM
5872 00002031 E8290A0000              CALL  NEC_OUTPUT
5873 00002036 B208                    MOV   DL,8                ; FILLER BYTE TO NEC
5874 00002038 E81C090000              CALL  GET_PARM
5875 0000203D E81D0A0000              CALL  NEC_OUTPUT
5876 00002042 58                      POP   eAX                 ; THROW AWAY ERROR
5877 00002043 E811070000              CALL  NEC_TERM            ; TERMINATE, RECEIVE STATUS, ETC,
5878                                  FM_DON:
5879 00002048 E8F8020000              CALL  XLAT_OLD            ; TRANSLATE STATE TO COMPATIBLE MODE
5880 0000204D E833080000              CALL  SETUP_END           ; VARIOUS CLEANUPS
5881 00002052 6689F3                  MOV   BX,SI               ; GET SAVED AL TO BL
5882 00002055 88D8                    MOV   AL,BL               ; PUT BACK FOR RETURN
5883 00002057 C3                      RETn
5884
5885                                  ;--------------------------------------------------------------------------
5886                                  ; FNC_ERR
5887                                  ;     INVALID FUNCTION REQUESTED OR INVALID DRIVE:
5888                                  ;     SET BAD COMMAND IN STATUS.
5889                                  ;
5890                                  ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
5891                                  ;--------------------------------------------------------------------------
5892                                  FNC_ERR:                        ; INVALID FUNCTION REQUEST
5893 00002058 6689F0                  MOV   AX,SI               ; RESTORE AL
5894 0000205B B401                    MOV   AH,BAD_CMD          ; SET BAD COMMAND ERROR
5895 0000205D 8825[C51D0000]          MOV   [DSKETTE_STATUS],AH ; STORE IN DATA AREA
5896 00002063 F9                      STC                       ; SET CARRY INDICATING ERROR
5897 00002064 C3                      RETn
5898
5899                                  ;--------------------------------------------------------------------------
```

```
                                ; DISK_PARMS (AH = 08H)
                                ;     READ DRIVE PARAMETERS.
                                ;
                                ; ON ENTRY:  DI : DRIVE #
                                ;
                                ; ON EXIT:   CL/[BP]   = BITS 7 & 6 HI 2 BITS OF MAX CYLINDER
                                ;                       BITS 0-5 MAX SECTORS/TRACK
                                ;            CH/[BP+1] = LOW 8 BITS OF MAX CYLINDER
                                ;            BL/[BP+2] = BITS 7-4 = 0
                                ;                       BITS 3-0 = VALID CMOS DRIVE TYPE
                                ;            BH/[BP+3] = 0
                                ;            DL/[BP+4] = # DRIVES INSTALLED (VALUE CHECKED)
                                ;            DH/[BP+5] = MAX HEAD #
                                ;            DI/[BP+6] = OFFSET TO DISK_BASE
                                ;            ES        = SEGMENT OF DISK_BASE
                                ;            AX        = 0
                                ;
                                ;            NOTE : THE ABOVE INFORMATION IS STORED IN THE USERS STACK AT
                                ;                   THE LOCATIONS WHERE THE MAIN ROUTINE WILL POP THEM
                                ;                   INTO THE APPROPRIATE REGISTERS BEFORE RETURNING TO THE
                                ;                   CALLER.
                                ;--------------------------------------------------------------------------
                                DSK_PARMS:
00002065 E8AA020000                   CALL   XLAT_NEW           ; TRANSLATE STATE TO PRESENT ARCH,
                                     ;       MOV    WORD [BP+2],0       ; DRIVE TYPE = 0
0000206A 29D2                         sub     edx, edx ; 20/02/2015
0000206C 895504                         mov  [ebp+4], edx ; 20/02/2015
                                     ;  MOV     AX, [EQUIP_FLAG]        ; LOAD EQUIPMENT FLAG FOR # DISKETTES
                                     ;  AND     AL,11000001B            ; KEEP DISKETTE DRIVE BITS
                                     ;  MOV     DL,2                    ; DISKETTE DRIVES = 2
                                     ;  CMP     AL,01000001B            ; 2 DRIVES INSTALLED ?
                                     ;  JZ      short STO_DL            ; IF YES JUMP
                                     ;  DEC     DL                      ; DISKETTE DRIVES = 1
                                     ;  CMP     AL,00000001B            ; 1 DRIVE INSTALLED ?
                                     ;  JNZ     short NON_DRV           ; IF NO JUMP
                                      ;sub   edx, edx
0000206F 66A1[16490000]               mov     ax, [fd0_type]
00002075 6621C0                       and     ax, ax
00002078 7474                         jz      short NON_DRV
0000207A FEC2                         inc     dl
0000207C 20E4                         and     ah, ah
0000207E 7402                         jz      short STO_DL
00002080 FEC2                         inc     dl
                                STO_DL:
                                      ;MOV   [BP+4],DL           ; STORE NUMBER OF DRIVES
00002082 895508                       mov    [ebp+8], edx ; 20/02/2015
00002085 6683FF01                     CMP    DI,1               ; CHECK FOR VALID DRIVE
00002089 7766                         JA     short NON_DRV1           ; DRIVE INVALID
                                      ;MOV   BYTE [BP+5],1      ; MAXIMUM HEAD NUMBER =   1
```

```
5949 0000208B C6450901                  mov   byte [ebp+9], 1  ; 20/02/2015
5950 0000208F E8BC080000                CALL  CMOS_TYPE          ; RETURN DRIVE TYPE IN AL
5951                                    ;;20/02/2015
5952                                    ;;JC  short CHK_EST      ; IF CMOS BAD CHECKSUM ESTABLISHED
5953                                    ;;OR  AL,AL              ; TEST FOR NO DRIVE TYPE
5954 00002094 7412                      JZ    short CHK_EST      ; JUMP IF SO
5955 00002096 E805020000                CALL  DR_TYPE_CHECK      ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
5956 0000209B 720B                      JC    short CHK_EST      ; TYPE NOT IN TABLE (POSSIBLE BAD CMOS)
5957                                    ;MOV  [BP+2],AL          ; STORE VALID CMOS DRIVE TYPE
5958 0000209D 884504                      mov [ebp+4], al ; 06/02/2015
5959 000020A0 8A4B04                    MOV    CL, [eBX+MD.SEC_TRK]     ; GET SECTOR/TRACK
5960 000020A3 8A6B0B                      MOV    CH, [eBX+MD.MAX_TRK]     ; GET MAX. TRACK NUMBER
5961 000020A6 EB36                      JMP   SHORT STO_CX       ; CMOS GOOD, USE CMOS
5962                                CHK_EST:
5963 000020A8 8AA7[D31D0000]            MOV   AH, [DSK_STATE+eDI] ; LOAD STATE FOR THIS DRIVE
5964 000020AE F6C410                    TEST  AH,MED_DET         ; CHECK FOR ESTABLISHED STATE
5965 000020B1 743E                      JZ    short NON_DRV1           ; CMOS BAD/INVALID OR UNESTABLISHED
5966                                USE_EST:
5967 000020B3 80E4C0                    AND   AH,RATE_MSK        ; ISOLATE STATE
5968 000020B6 80FC80                    CMP   AH,RATE_250        ; RATE 250 ?
5969 000020B9 7557                      JNE   short USE_EST2           ; NO, GO CHECK OTHER RATE
5970
5971                                ;-----      DATA RATE IS 250 KBS, TRY 360 KB TABLE FIRST
5972
5973 000020BB B001                      MOV   AL,01              ; DRIVE TYPE 1 (360KB)
5974 000020BD E8DE010000                CALL  DR_TYPE_CHECK      ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
5975 000020C2 8A4B04                      MOV    CL, [eBX+MD.SEC_TRK]    ; GET SECTOR/TRACK
5976 000020C5 8A6B0B                      MOV    CH, [eBX+MD.MAX_TRK]    ; GET MAX. TRACK NUMBER
5977 000020C8 F687[D31D0000]01          TEST  byte [DSK_STATE+eDI],TRK_CAPA ; 80 TRACK ?
5978 000020CF 740D                      JZ    short STO_CX       ; MUST BE 360KB DRIVE
5979
5980                                ;-----      IT IS 1.44 MB DRIVE
5981
5982                                PARM144:
5983 000020D1 B004                      MOV   AL,04              ; DRIVE TYPE 4 (1.44MB)
5984 000020D3 E8C8010000                CALL  DR_TYPE_CHECK      ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
5985 000020D8 8A4B04                      MOV    CL, [eBX+MD.SEC_TRK]    ; GET SECTOR/TRACK
5986 000020DB 8A6B0B                      MOV    CH, [eBX+MD.MAX_TRK]    ; GET MAX. TRACK NUMBER
5987                                STO_CX:
5988 000020DE 894D00                    MOV   [eBP],eCX          ; SAVE POINTER IN STACK FOR RETURN
5989                                ES_DI:
5990                                    ;MOV  [BP+6],BX          ; ADDRESS OF MEDIA/DRIVE PARM TABLE
5991 000020E1 895D0C                    mov   [ebp+12], ebx ; 06/02/2015
5992                                    ;MOV  AX,CS              ; SEGMENT MEDIA/DRIVE PARAMETER TABLE
5993                                    ;MOV  ES,AX              ; ES IS SEGMENT OF TABLE
5994                                DP_OUT:
5995 000020E4 E85C020000                CALL  XLAT_OLD           ; TRANSLATE STATE TO COMPATIBLE MODE
5996 000020E9 6631C0                    XOR   AX,AX              ; CLEAR
5997 000020EC F8                        CLC
```

```
5998 000020ED C3                        RETn
5999
6000                                ;-----      NO DRIYE PRESENT HANDLER
6001
6002                                NON_DRV:
6003                                    ;MOV   BYTE [BP+4],0       ; CLEAR NUMBER OF DRIVES
6004 000020EE 895508                    mov    [ebp+8], edx ; 0 ; 20/02/2015
6005                                NON_DRV1:
6006 000020F1 6681FF8000                CMP    DI,80H              ; CHECK FOR FIXED MEDIA TYPE REQUEST
6007 000020F6 720C                      JB     short NON_DRV2            ; CONTINUE IF NOT REQUEST FALL THROUGH
6008
6009                                ;-----      FIXED DISK REQUEST FALL THROUGH ERROR
6010
6011 000020F8 E848020000                CALL   XLAT_OLD            ; ELSE TRANSLATE TO COMPATIBLE MODE
6012 000020FD 6689F0                    MOV    AX,SI               ; RESTORE AL
6013 00002100 B401                      MOV    AH,BAD_CMD          ; SET BAD COMMAND ERROR
6014 00002102 F9                        STC
6015 00002103 C3                        RETn
6016
6017                                NON_DRV2:
6018                                    ;XOR   AX,AX               ; CLEAR PARMS IF NO DRIVES OR CMOS BAD
6019 00002104 31C0                      xor    eax, eax
6020 00002106 66894500                  MOV    [eBP],AX            ; TRACKS, SECTORS/TRACK = 0
6021                                    ;MOV   [BP+5],AH           ; HEAD = 0
6022 0000210A 886509                    mov    [ebp+9], ah ; 06/02/2015
6023                                    ;MOV   [BP+6],AX           ; OFFSET TO DISK_BASE = 0
6024 0000210D 89450C                    mov    [ebp+12], eax
6025                                    ;MOV   ES,AX               ; ES IS SEGMENT OF TABLE
6026 00002110 EBD2                      JMP    SHORT DP_OUT
6027
6028                                ;-----      DATA RATE IS EITHER 300 KBS OR 500 KBS, TRY 1.2 MB TABLE FIRST
6029
6030                                USE_EST2:
6031 00002112 B002                      MOV    AL,02               ; DRIVE TYPE 2 (1.2MB)
6032 00002114 E887010000                CALL   DR_TYPE_CHECK       ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
6033 00002119 8A4B04                      MOV     CL, [eBX+MD.SEC_TRK]    ; GET SECTOR/TRACK
6034 0000211C 8A6B0B                      MOV     CH, [eBX+MD.MAX_TRK]    ; GET MAX. TRACK NUMBER
6035 0000211F 80FC40                    CMP    AH,RATE_300         ; RATE 300 ?
6036 00002122 74BA                      JZ     short STO_CX        ; MUST BE 1.2MB DRIVE
6037 00002124 EBAB                      JMP    SHORT PARM144       ; ELSE, IT IS 1.44MB DRIVE
6038
6039                                ;--------------------------------------------------------------------------
6040                                ; DISK_TYPE (AH = 15H)
6041                                ;     THIS ROUTINE RETURNS THE TYPE OF MEDIA INSTALLED.
6042                                ;
6043                                ;  ON ENTRY: DI = DRIVE #
6044                                ;
6045                                ;  ON EXIT:  AH = DRIVE TYPE, CY=0
6046                                ;--------------------------------------------------------------------------
```

```
6047                                    DSK_TYPE:
6048 00002126 E8E9010000                      CALL   XLAT_NEW           ; TRANSLATE STATE TO PRESENT ARCH.
6049 0000212B 8A87[D31D0000]                  MOV    AL, [DSK_STATE+eDI] ; GET PRESENT STATE INFORMATION
6050 00002131 08C0                            OR     AL,AL              ; CHECK FOR NO DRIVE
6051 00002133 7418                            JZ     short NO_DRV
6052 00002135 B401                            MOV    AH,NOCHGLN         ; NO CHANGE LINE FOR 40 TRACK DRIVE
6053 00002137 A801                            TEST   AL,TRK_CAPA        ; IS THIS DRIVE AN 80 TRACK DRIVE?
6054 00002139 7402                            JZ     short DT_BACK            ; IF NO JUMP
6055 0000213B B402                            MOV    AH,CHGLN           ; CHANGE LINE FOR 80 TRACK DRIVE
6056                                    DT_BACK:
6057 0000213D 6650                            PUSH   AX                 ; SAVE RETURN VALUE
6058 0000213F E801020000                      CALL   XLAT_OLD           ; TRANSLATE STATE TO COMPATIBLE MODE
6059 00002144 6658                            POP    AX                 ; RESTORE RETURN VALUE
6060 00002146 F8                              CLC                       ; NO ERROR
6061 00002147 6689F3                          MOV    BX,SI              ; GET SAVED AL TO BL
6062 0000214A 88D8                            MOV    AL,BL              ; PUT BACK FOR RETURN
6063 0000214C C3                              RETn
6064                                    NO_DRV:
6065 0000214D 30E4                            XOR    AH,AH              ; NO DRIVE PRESENT OR UNKNOWN
6066 0000214F EBEC                            JMP    SHORT DT_BACK
6067
6068                                    ;--------------------------------------------------------------------------
6069                                    ; DISK_CHANGE      (AH = 16H)
6070                                    ;     THIS ROUTINE RETURNS THE STATE OF THE DISK CHANGE LINE.
6071                                    ;
6072                                    ; ON ENTRY:  DI = DRIVE #
6073                                    ;
6074                                    ; ON EXIT:   AH = @DSKETTE_STATUS
6075                                    ;                 00 - DISK CHANGE LINE INACTIVE, CY = 0
6076                                    ;                 06 - DISK CHANGE LINE ACTIVE, CY = 1
6077                                    ;--------------------------------------------------------------------------
6078                                    DSK_CHANGE:
6079 00002151 E8BE010000                      CALL   XLAT_NEW           ; TRANSLATE STATE TO PRESENT ARCH.
6080 00002156 8A87[D31D0000]                  MOV    AL, [DSK_STATE+eDI] ; GET MEDIA STATE INFORMATION
6081 0000215C 08C0                            OR     AL,AL              ; DRIVE PRESENT ?
6082 0000215E 7422                            JZ     short DC_NON       ; JUMP IF NO DRIVE
6083 00002160 A801                            TEST   AL,TRK_CAPA        ; 80 TRACK DRIVE ?
6084 00002162 7407                            JZ     short SETIT        ; IF SO , CHECK CHANGE LINE
6085                                    DC0:
6086 00002164 E8830A0000                        CALL    READ_DSKCHNG           ; GO CHECK STATE OF DISK CHANGE LINE
6087 00002169 7407                            JZ     short FINIS        ; CHANGE LINE NOT ACTIVE
6088
6089 0000216B C605[C51D0000]06          SETIT:      MOV    byte [DSKETTE_STATUS], MEDIA_CHANGE ; INDICATE MEDIA REMOVED
6090
6091 00002172 E8CE010000                FINIS:      CALL   XLAT_OLD           ; TRANSLATE STATE TO COMPATIBLE MODE
6092 00002177 E809070000                      CALL   SETUP_END          ; VARIOUS CLEANUPS
6093 0000217C 6689F3                          MOV    BX,SI              ; GET SAVED AL TO BL
6094 0000217F 88D8                            MOV    AL,BL              ; PUT BACK FOR RETURN
6095 00002181 C3                              RETn
```

```
6096                                    DC_NON:
6097 00002182 800D[C51D0000]80                  OR    byte [DSKETTE_STATUS], TIME_OUT ; SET TIMEOUT, NO DRIVE
6098 00002189 EBE7                              JMP   SHORT FINIS
6099
6100                                    ;--------------------------------------------------------------------------
6101                                    ; FORMAT_SET (AH = 17H)
6102                                    ;     THIS ROUTINE IS USED TO ESTABLISH THE TYPE OF MEDIA TO BE USED
6103                                    ;     FOR THE FOLLOWING FORMAT OPERATION.
6104                                    ;
6105                                    ; ON ENTRY:  SI LOW = DASD TYPE FOR FORMAT
6106                                    ;            DI     = DRIVE #
6107                                    ;
6108                                    ; ON EXIT:   @DSKETTE_STATUS REFLECTS STATUS
6109                                    ;            AH = @DSKETTE_STATUS
6110                                    ;            CY = 1 IF ERROR
6111                                    ;--------------------------------------------------------------------------
6112                                    FORMAT_SET:
6113 0000218B E884010000                        CALL  XLAT_NEW            ; TRANSLATE STATE TO PRESENT ARCH.
6114 00002190 6656                              PUSH  SI                  ; SAVE DASD TYPE
6115 00002192 6689F0                            MOV   AX,SI               ; AH = ? , AL , DASD TYPE
6116 00002195 30E4                              XOR   AH,AH               ; AH , 0 , AL , DASD TYPE
6117 00002197 6689C6                            MOV   SI,AX               ; SI = DASD TYPE
6118 0000219A 80A7[D31D0000]0F                  AND   byte [DSK_STATE+eDI], ~(MED_DET+DBL_STEP+RATE_MSK) ; CLEAR STATE
6119 000021A1 664E                              DEC   SI                  ; CHECK FOR 320/360K MEDIA & DRIVE
6120 000021A3 7509                              JNZ   short NOT_320       ; BYPASS IF NOT
6121 000021A5 808F[D31D0000]90                  OR    byte [DSK_STATE+eDI], MED_DET+RATE_250 ; SET TO 320/360
6122 000021AC EB48                              JMP   SHORT S0
6123
6124                                    NOT_320:
6125 000021AE E8B6030000                        CALL  MED_CHANGE          ; CHECK FOR TIME_OUT
6126 000021B3 803D[C51D0000]80                  CMP   byte [DSKETTE_STATUS], TIME_OUT
6127 000021BA 743A                              JZ    short S0            ; IF TIME OUT TELL CALLER
6128                                    S3:
6129 000021BC 664E                              DEC   SI                  ; CHECK FOR 320/360K IN 1.2M DRIVE
6130 000021BE 7509                              JNZ   short NOT_320_12    ; BYPASS IF NOT
6131 000021C0 808F[D31D0000]70                  OR    byte [DSK_STATE+eDI], MED_DET+DBL_STEP+RATE_300 ; SET STATE
6132 000021C7 EB2D                              JMP   SHORT S0
6133
6134                                    NOT_320_12:
6135 000021C9 664E                              DEC   SI                  ; CHECK FOR 1.2M MEDIA IN 1.2M DRIVE
6136 000021CB 7509                              JNZ   short NOT_12        ; BYPASS IF NOT
6137 000021CD 808F[D31D0000]10                  OR    byte [DSK_STATE+eDI], MED_DET+RATE_500 ; SET STATE VARIABLE
6138 000021D4 EB20                              JMP   SHORT S0            ; RETURN TO CALLER
6139
6140                                    NOT_12:
6141 000021D6 664E                              DEC   SI                  ; CHECK FOR SET DASD TYPE 04
6142 000021D8 752B                              JNZ   short FS_ERR        ; BAD COMMAND EXIT IF NOT VALID TYPE
6143
6144 000021DA F687[D31D0000]04                  TEST  byte [DSK_STATE+eDI], DRV_DET ; DRIVE DETERMINED ?
```

```
6145 000021E1 740B                      JZ     short ASSUME        ; IF STILL NOT DETERMINED ASSUME
6146 000021E3 B050                      MOV    AL,MED_DET+RATE_300
6147 000021E5 F687[D31D0000]02            TEST    byte [DSK_STATE+eDI], FMT_CAPA ; MULTIPLE FORMAT CAPABILITY ?
6148 000021EC 7502                      JNZ    short OR_IT_IN            ; IF 1.2 M THEN DATA RATE 300
6149
6150                                 ASSUME:
6151 000021EE B090                      MOV    AL,MED_DET+RATE_250 ; SET UP
6152
6153                                 OR_IT_IN:
6154 000021F0 0887[D31D0000]            OR     byte [DSK_STATE+eDI], AL  ; OR IN THE CORRECT STATE
6155                                 S0:
6156 000021F6 E84A010000                CALL   XLAT_OLD            ; TRANSLATE STATE TO COMPATIBLE MODE
6157 000021FB E885060000                CALL   SETUP_END           ; VARIOUS CLEANUPS
6158 00002200 665B                      POP    BX                  ; GET SAVED AL TO BL
6159 00002202 88D8                      MOV    AL,BL               ; PUT BACK FOR RETURN
6160 00002204 C3                        RETn
6161
6162                                 FS_ERR:
6163 00002205 C605[C51D0000]01          MOV    byte [DSKETTE_STATUS], BAD_CMD ; UNKNOWN STATE,BAD COMMAND
6164 0000220C EBE8                      JMP    SHORT S0
6165
6166                                 ;--------------------------------------------------------------------------------
6167                                 ; SET_MEDIA  (AH = 18H)
6168                                 ;     THIS ROUTINE SETS THE TYPE OF MEDIA AND DATA RATE
6169                                 ;     TO BE USED FOR THE FOLLOWING FORMAT OPERATION.
6170                                 ;
6171                                 ; ON ENTRY:
6172                                 ;     [BP]   = SECTOR PER TRACK
6173                                 ;     [BP+1] = TRACK #
6174                                 ;     DI     = DRIVE #
6175                                 ;
6176                                 ; ON EXIT:
6177                                 ;     @DSKETTE_STATUS REFLECTS STATUS
6178                                 ;     IF NO ERROR:
6179                                 ;           AH = 0
6180                                 ;           CY = 0
6181                                 ;           ES = SEGMENT OF MEDIA/DRIVE PARAMETER TABLE
6182                                 ;           DI/[BP+6] = OFFSET OF MEDIA/DRIVE PARAMETER TABLE
6183                                 ;     IF ERROR:
6184                                 ;           AH = @DSKETTE_STATUS
6185                                 ;           CY = 1
6186                                 ;--------------------------------------------------------------------------------
6187                                 SET_MEDIA:
6188 0000220E E801010000                CALL   XLAT_NEW            ; TRANSLATE STATE TO PRESENT ARCH.
6189 00002213 F687[D31D0000]01            TEST    byte [DSK_STATE+eDI], TRK_CAPA ; CHECK FOR CHANGE LINE AVAILABLE
6190 0000221A 7415                      JZ     short SM_CMOS       ; JUMP IF 40 TRACK DRIVE
6191 0000221C E848030000                CALL   MED_CHANGE          ; RESET CHANGE LINE
6192 00002221 803D[C51D0000]80          CMP    byte [DSKETTE_STATUS], TIME_OUT ; IF TIME OUT TELL CALLER
6193 00002228 746B                      JE     short SM_RTN
```

```
6194 0000222A C605[C51D0000]00          MOV   byte [DSKETTE_STATUS], 0 ; CLEAR STATUS
6195                                    SM_CMOS:
6196 00002231 E81A070000                CALL  CMOS_TYPE           ; RETURN DRIVE TYPE IN (AL)
6197                                    ;;20/02/2015
6198                                    ;;JC  short MD_NOT_FND    ; ERROR IN CMOS
6199                                    ;;OR  AL,AL               ; TEST FOR NO DRIVE
6200 00002236 745D                      JZ    short SM_RTN        ; RETURN IF SO
6201 00002238 E863000000                CALL  DR_TYPE_CHECK       ; RTN CS:BX = MEDIA/DRIVE PARAM TBL
6202 0000223D 7231                      JC    short MD_NOT_FND    ; TYPE NOT IN TABLE (BAD CMOS)
6203 0000223F 57                        PUSH  eDI                 ; SAVE REG.
6204 00002240 31DB                      XOR   eBX,eBX                   ; BX = INDEX TO DR. TYPE TABLE
6205 00002242 B906000000                MOV   eCX,DR_CNT          ; CX = LOOP COUNT
6206                                    DR_SEARCH:
6207 00002247 8AA3[DB1D0000]            MOV   AH, [DR_TYPE+eBX]   ; GET DRIVE TYPE
6208 0000224D 80E47F                    AND   AH,BIT7OFF          ; MASK OUT MSB
6209 00002250 38E0                      CMP   AL,AH               ; DRIVE TYPE MATCH ?
6210 00002252 7516                      JNE   short NXT_MD        ; NO, CHECK NEXT DRIVE TYPE
6211                                    DR_FND:
6212 00002254 8BBB[DC1D0000]            MOV   eDI, [DR_TYPE+eBX+1]      ; DI = MEDIA/DRIVE PARAM TABLE
6213                                    MD_SEARCH:
6214 0000225A 8A6704                      MOV   AH, [eDI+MD.SEC_TRK]    ; GET SECTOR/TRACK
6215 0000225D 386500                    CMP   [eBP],AH            ; MATCH?
6216 00002260 7508                      JNE   short NXT_MD        ; NO, CHECK NEXT MEDIA
6217 00002262 8A670B                      MOV   AH, [eDI+MD.MAX_TRK]    ; GET MAX. TRACK #
6218 00002265 386501                    CMP   [eBP+1],AH          ; MATCH?
6219 00002268 740F                      JE    short MD_FND        ; YES, GO GET RATE
6220                                    NXT_MD:
6221                                    ;ADD   BX,3               ; CHECK NEXT DRIVE TYPE
6222 0000226A 83C305                      add  ebx, 5 ; 18/02/2015
6223 0000226D E2D8                      LOOP   DR_SEARCH
6224 0000226F 5F                        POP   eDI                 ; RESTORE REG.
6225                                    MD_NOT_FND:
6226 00002270 C605[C51D0000]0C          MOV   byte [DSKETTE_STATUS], MED_NOT_FND ; ERROR, MEDIA TYPE NOT FOUND
6227 00002277 EB1C                      JMP   SHORT SM_RTN        ; RETURN
6228                                    MD_FND:
6229 00002279 8A470C                      MOV   AL, [eDI+MD.RATE]       ; GET RATE
6230 0000227C 3C40                      CMP   AL,RATE_300         ; DOUBLE STEP REQUIRED FOR RATE 300
6231 0000227E 7502                      JNE   short MD_SET
6232 00002280 0C20                      OR    AL,DBL_STEP
6233                                    MD_SET:
6234                                    ;MOV   [BP+6],DI          ; SAVE TABLE POINTER IN STACK
6235 00002282 897D0C                    mov   [ebp+12], edi ; 18/02/2015
6236 00002285 0C10                      OR    AL,MED_DET          ; SET MEDIA ESTABLISHED
6237 00002287 5F                        POP   eDI
6238 00002288 80A7[D31D0000]0F          AND   byte [DSK_STATE+eDI], ~(MED_DET+DBL_STEP+RATE_MSK) ; CLEAR STATE
6239 0000228F 0887[D31D0000]            OR    byte [DSK_STATE+eDI], AL
6240                                    ;MOV   AX, CS             ; SEGMENT OF MEDIA/DRIVE PARAMETER TABLE
6241                                    ;MOV   ES, AX             ; ES IS SEGMENT OF TABLE
6242                                    SM_RTN:
```

```
6243 00002295 E8AB000000                CALL   XLAT_OLD           ; TRANSLATE STATE TO COMPATIBLE MODE
6244 0000229A E8E6050000                CALL   SETUP_END          ; VARIOUS CLEANUPS
6245 0000229F C3                        RETn
6246
6247                                ;--------------------------------------------------------
6248                                ; DR_TYPE_CHECK                                        :
6249                                ;     CHECK IF THE GIVEN DRIVE TYPE IN REGISTER (AL)           :
6250                                ;     IS SUPPORTED IN BIOS DRIVE TYPE TABLE            :
6251                                ; ON ENTRY:                                          :
6252                                ;     AL = DRIVE TYPE                                  :
6253                                ; ON EXIT:                                           :
6254                                ;     CS = SEGMENT MEDIA/DRIVE PARAMETER TABLE (CODE)          :
6255                                ;     CY = 0       DRIVE TYPE SUPPORTED                :
6256                                ;          BX = OFFSET TO MEDIA/DRIVE PARAMETER TABLE  :
6257                                ;     CY = 1 DRIVE TYPE NOT SUPPORTED                  :
6258                                ; REGISTERS ALTERED: BX                                :
6259                                ;--------------------------------------------------------
6260                                DR_TYPE_CHECK:
6261 000022A0 6650                        PUSH   AX
6262 000022A2 51                          PUSH   eCX
6263 000022A3 31DB                        XOR    eBX,eBX                  ; BX = INDEX TO DR_TYPE TABLE
6264 000022A5 B906000000                  MOV    eCX,DR_CNT          ; CX = LOOP COUNT
6265                                TYPE_CHK:
6266 000022AA 8AA3[DB1D0000]              MOV    AH,[DR_TYPE+eBX]    ; GET DRIVE TYPE
6267 000022B0 38E0                        CMP    AL,AH               ; DRIVE TYPE MATCH?
6268 000022B2 740D                        JE     short DR_TYPE_VALID ; YES, RETURN WITH CARRY RESET
6269                                      ;ADD   BX,3                ; CHECK NEXT DRIVE TYPE
6270 000022B4 83C305                        add  ebx, 5 ; 16/02/2015 (32 bit address modification)
6271 000022B7 E2F1                        LOOP    TYPE_CHK
6272                                      ;
6273 000022B9 BB[3A1E0000]                mov    ebx, MD_TBL6        ; 1.44MB fd parameter table
6274                                                                  ; Default for GET_PARM (11/12/2014)
6275                                      ;
6276 000022BE F9                          STC                        ; DRIVE TYPE NOT FOUND IN TABLE
6277 000022BF EB06                        JMP    SHORT TYPE_RTN
6278                                DR_TYPE_VALID:
6279 000022C1 8B9B[DC1D0000]              MOV    eBX,[DR_TYPE+eBX+1]       ; BX = MEDIA TABLE
6280                                TYPE_RTN:
6281 000022C7 59                          POP    eCX
6282 000022C8 6658                        POP    AX
6283 000022CA C3                          RETn
6284
6285                                ;--------------------------------------------------------
6286                                ; SEND_SPEC                                            :
6287                                ;     SEND THE SPECIFY COMMAND TO CONTROLLER USING DATA FROM     :
6288                                ;     THE DRIVE PARAMETER TABLE POINTED BY @DISK_POINTER   :
6289                                ; ON ENTRY:  @DISK_POINTER = DRIVE PARAMETER TABLE         :
6290                                ; ON EXIT:   NONE                                      :
6291                                ; REGISTERS ALTERED: CX, DX                            :
```

```
6292                                    ;--------------------------------------------------------------
6293                                    SEND_SPEC:
6294 000022CB 50                              PUSH   eAX                ; SAVE AX
6295 000022CC B8[F2220000]                    MOV    eAX, SPECBAC       ; LOAD ERROR ADDRESS
6296 000022D1 50                              PUSH   eAX                ; PUSH NEC_OUT ERROR RETURN
6297 000022D2 B403                            MOV    AH,03H             ; SPECIFY COMMAND
6298 000022D4 E886070000                      CALL   NEC_OUTPUT         ; OUTPUT THE COMMAND
6299 000022D9 28D2                            SUB    DL,DL              ; FIRST SPECIFY BYTE
6300 000022DB E879060000                      CALL   GET_PARM           ; GET PARAMETER TO AH
6301 000022E0 E87A070000                      CALL   NEC_OUTPUT         ; OUTPUT THE COMMAND
6302 000022E5 B201                            MOV    DL,1               ; SECOND SPECIFY BYTE
6303 000022E7 E86D060000                      CALL   GET_PARM           ; GET PARAMETER TO AH
6304 000022EC E86E070000                      CALL   NEC_OUTPUT         ; OUTPUT THE COMMAND
6305 000022F1 58                              POP    eAX                ; POP ERROR RETURN
6306                                    SPECBAC:
6307 000022F2 58                              POP    eAX                ; RESTORE ORIGINAL AX VALUE
6308 000022F3 C3                              RETn
6309
6310                                    ;--------------------------------------------------------------
6311                                    ; SEND_SPEC_MD                                          :
6312                                    ;     SEND THE SPECIFY COMMAND TO CONTROLLER USING DATA FROM     :
6313                                    ;     THE MEDIA/DRIVE PARAMETER TABLE POINTED BY (CS:BX)   :
6314                                    ; ON ENTRY:  CS:BX = MEDIA/DRIVE PARAMETER TABLE          :
6315                                    ; ON EXIT:   NONE                                  :
6316                                    ; REGISTERS ALTERED: AX                                 :
6317                                    ;--------------------------------------------------------------
6318                                    SEND_SPEC_MD:
6319 000022F4 50                              PUSH   eAX                ; SAVE RATE DATA
6320 000022F5 B8[12230000]                    MOV    eAX, SPEC_ESBAC          ; LOAD ERROR ADDRESS
6321 000022FA 50                              PUSH   eAX                ; PUSH NEC_OUT ERROR RETURN
6322 000022FB B403                            MOV    AH,03H             ; SPECIFY COMMAND
6323 000022FD E85D070000                      CALL   NEC_OUTPUT         ; OUTPUT THE COMMAND
6324 00002302 8A23                              MOV     AH, [eBX+MD.SPEC1]      ; GET 1ST SPECIFY BYTE
6325 00002304 E856070000                      CALL   NEC_OUTPUT         ; OUTPUT THE COMMAND
6326 00002309 8A6301                            MOV     AH, [eBX+MD.SPEC2]      ; GET SECOND SPECIFY BYTE
6327 0000230C E84E070000                      CALL   NEC_OUTPUT         ; OUTPUT THE COMMAND
6328 00002311 58                              POP    eAX                ; POP ERROR RETURN
6329                                    SPEC_ESBAC:
6330 00002312 58                              POP    eAX                ; RESTORE ORIGINAL AX VALUE
6331 00002313 C3                              RETn
6332
6333                                    ;--------------------------------------------------------------------------------
6334                                    ; XLAT_NEW
6335                                    ;     TRANSLATES DISKETTE STATE LOCATIONS FROM COMPATIBLE
6336                                    ;     MODE TO NEW ARCHITECTURE.
6337                                    ;
6338                                    ; ON ENTRY:  DI = DRIVE #
6339                                    ;--------------------------------------------------------------------------------
6340                                    XLAT_NEW:
```

```
6341 00002314 83FF01                    CMP    eDI,1                     ; VALID DRIVE
6342 00002317 7725                      JA     short XN_OUT              ; IF INVALID BACK
6343 00002319 80BF[D31D0000]00          CMP    byte [DSK_STATE+eDI], 0          ; NO DRIVE ?
6344 00002320 741D                      JZ     short DO_DET              ; IF NO DRIVE ATTEMPT DETERMINE
6345 00002322 6689F9                    MOV    CX,DI                     ; CX = DRIVE NUMBER
6346 00002325 C0E102                    SHL    CL,2                      ; CL = SHIFT COUNT, A=0, B=4
6347 00002328 A0[B52C0000]              MOV    AL, [HF_CNTRL]                 ; DRIVE INFORMATION
6348 0000232D D2C8                      ROR    AL,CL                     ; TO LOW NIBBLE
6349 0000232F 2407                      AND    AL,DRV_DET+FMT_CAPA+TRK_CAPA     ; KEEP DRIVE BITS
6350 00002331 80A7[D31D0000]F8            AND     byte [DSK_STATE+eDI], ~(DRV_DET+FMT_CAPA+TRK_CAPA)
6351 00002338 0887[D31D0000]            OR     [DSK_STATE+eDI], AL       ; UPDATE DRIVE STATE
6352                                 XN_OUT:
6353 0000233E C3                        RETn
6354                                 DO_DET:
6355 0000233F E8B5080000                CALL   DRIVE_DET                 ; TRY TO DETERMINE
6356 00002344 C3                        RETn
6357
6358                                 ;--------------------------------------------------------------------------------
6359                                 ; XLAT_OLD
6360                                 ;     TRANSLATES DISKETTE STATE LOCATIONS FROM NEW
6361                                 ;     ARCHITECTURE TO COMPATIBLE MODE.
6362                                 ;
6363                                 ; ON ENTRY:  DI = DRIVE
6364                                 ;--------------------------------------------------------------------------------
6365                                 XLAT_OLD:
6366 00002345 83FF01                    CMP    eDI,1              ; VALID DRIVE ?
6367                                      ;JA     short XO_OUT             ; IF INVALID BACK
6368 00002348 0F8786000000               ja     XO_OUT
6369 0000234E 80BF[D31D0000]00           CMP  byte [DSK_STATE+eDI],0     ; NO DRIVE ?
6370 00002355 747D                      JZ     short XO_OUT       ; IF NO DRIVE TRANSLATE DONE
6371
6372                                 ;-----      TEST FOR SAVED DRIVE INFORMATION ALREADY SET
6373
6374 00002357 6689F9                    MOV    CX,DI              ; CX = DRIVE NUMBER
6375 0000235A C0E102                    SHL    CL,2               ; CL = SHIFT COUNT, A=0, B=4
6376 0000235D B402                      MOV    AH,FMT_CAPA        ; LOAD MULTIPLE DATA RATE BIT MASK
6377 0000235F D2CC                      ROR    AH,CL              ; ROTATE BY MASK
6378 00002361 8425[B52C0000]            TEST   [HF_CNTRL], AH           ; MULTIPLE-DATA RATE DETERMINED ?
6379 00002367 751C                      JNZ    short SAVE_SET           ; IF SO, NO NEED TO RE-SAVE
6380
6381                                 ;-----      ERASE DRIVE BITS IN @HF_CNTRL FOR THIS DRIVE
6382
6383 00002369 B407                      MOV    AH,DRV_DET+FMT_CAPA+TRK_CAPA ; MASK TO KEEP
6384 0000236B D2CC                      ROR    AH,CL              ; FIX MASK TO KEEP
6385 0000236D F6D4                      NOT    AH                 ; TRANSLATE MASK
6386 0000236F 2025[B52C0000]            AND    [HF_CNTRL], AH           ; KEEP BITS FROM OTHER DRIVE INTACT
6387
6388                                 ;-----      ACCESS CURRENT DRIVE BITS AND STORE IN @HF_CNTRL
6389
```

```
6390 00002375 8A87[D31D0000]              MOV    AL, [DSK_STATE+eDI] ; ACCESS STATE
6391 0000237B 2407                        AND    AL,DRV_DET+FMT_CAPA+TRK_CAPA ; KEEP DRIVE BITS
6392 0000237D D2C8                        ROR    AL,CL               ; FIX FOR THIS DRIVE
6393 0000237F 0805[B52C0000]              OR     [HF_CNTRL], AL            ; UPDATE SAVED DRIVE STATE
6394
6395                                   ;-----      TRANSLATE TO COMPATIBILITY MODE
6396
6397                                   SAVE_SET:
6398 00002385 8AA7[D31D0000]              MOV    AH, [DSK_STATE+eDI] ; ACCESS STATE
6399 0000238B 88E7                        MOV    BH,AH               ; TO BH FOR LATER
6400 0000238D 80E4C0                      AND    AH,RATE_MSK         ; KEEP ONLY RATE
6401 00002390 80FC00                      CMP    AH,RATE_500         ; RATE 500 ?
6402 00002393 7410                        JZ     short CHK_144       ; YES 1.2/1.2 OR 1.44/1.44
6403 00002395 B001                        MOV    AL,M3D1U            ; AL = 360 IN 1.2 UNESTABLISHED
6404 00002397 80FC40                      CMP    AH,RATE_300         ; RATE 300 ?
6405 0000239A 7518                        JNZ    short CHK_250       ; NO, 360/360, 720/720 OR 720/1.44
6406 0000239C F6C720                      TEST   BH,DBL_STEP         ; CHECK FOR DOUBLE STEP
6407 0000239F 751F                        JNZ    short TST_DET       ; MUST BE 360 IN 1.2
6408                                   UNKNO:
6409 000023A1 B007                        MOV    AL,MED_UNK          ; NONE OF THE ABOVE
6410 000023A3 EB22                        JMP    SHORT AL_SET        ; PROCESS COMPLETE
6411                                   CHK_144:
6412 000023A5 E8A6050000                  CALL   CMOS_TYPE           ; RETURN DRIVE TYPE IN (AL)
6413                                      ;;20/02/2015
6414                                      ;;JC   short UNKNO         ; ERROR, SET 'NONE OF ABOVE'
6415 000023AA 74F5                        jz     short UNKNO ;; 20/02/2015
6416 000023AC 3C02                        CMP    AL,2                ; 1.2MB DRIVE ?
6417 000023AE 75F1                        JNE    short UNKNO         ; NO, GO SET 'NONE OF ABOVE'
6418 000023B0 B002                        MOV    AL,M1D1U            ; AL = 1.2 IN 1.2 UNESTABLISHED
6419 000023B2 EB0C                        JMP    SHORT TST_DET
6420                                   CHK_250:
6421 000023B4 B000                        MOV    AL,M3D3U            ; AL = 360 IN 360 UNESTABLISHED
6422 000023B6 80FC80                      CMP    AH,RATE_250         ; RATE 250 ?
6423 000023B9 75E6                        JNZ    short UNKNO         ; IF SO FALL IHRU
6424 000023BB F6C701                      TEST   BH,TRK_CAPA         ; 80 TRACK CAPABILITY ?
6425 000023BE 75E1                        JNZ    short UNKNO         ; IF SO JUMP, FALL THRU TEST DET
6426                                   TST_DET:
6427 000023C0 F6C710                      TEST   BH,MED_DET          ; DETERMINED ?
6428 000023C3 7402                        JZ     short AL_SET        ; IF NOT THEN SET
6429 000023C5 0403                        ADD    AL,3                ; MAKE DETERMINED/ESTABLISHED
6430                                   AL_SET:
6431 000023C7 80A7[D31D0000]F8            AND    byte [DSK_STATE+eDI], ~(DRV_DET+FMT_CAPA+TRK_CAPA) ; CLEAR DRIVE
6432 000023CE 0887[D31D0000]              OR     [DSK_STATE+eDI], AL ; REPLACE WITH COMPATIBLE MODE
6433                                   XO_OUT:
6434 000023D4 C3                          RETn
6435
6436                                   ;--------------------------------------------------------------------------------
6437                                   ; RD_WR_VF
6438                                   ;     COMMON READ, WRITE AND VERIFY:
```

```
6439                                    ;     MAIN LOOP FOR STATE RETRIES.
6440                                    ;
6441                                    ; ON ENTRY:  AH = READ/WRITE/VERIFY NEC PARAMETER
6442                                    ;            AL = READ/WRITE/VERIFY DMA PARAMETER
6443                                    ;
6444                                    ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
6445                                    ;--------------------------------------------------------------------------------
6446                                    RD_WR_VF:
6447 000023D5 6650                             PUSH   AX                  ; SAVE DMA, NEC PARAMETERS
6448 000023D7 E838FFFFFF                       CALL   XLAT_NEW            ; TRANSLATE STATE TO PRESENT ARCH.
6449 000023DC E8F3000000                       CALL   SETUP_STATE         ; INITIALIZE START AND END RATE
6450 000023E1 6658                             POP    AX                  ; RESTORE READ/WRITE/VERIFY
6451                                    DO_AGAIN:
6452 000023E3 6650                             PUSH   AX                  ; SAVE READ/WRITE/VERIFY PARAMETER
6453 000023E5 E87F010000                       CALL   MED_CHANGE          ; MEDIA CHANGE AND RESET IF CHANGED
6454 000023EA 6658                             POP    AX                  ; RESTORE READ/WRITE/VERIFY
6455 000023EC 0F82C9000000                       JC      RWV_END                ; MEDIA CHANGE ERROR OR TIME-OUT
6456                                    RWV:
6457 000023F2 6650                             PUSH   AX                  ; SAVE READ/WRITE/VERIFY PARAMETER
6458 000023F4 8AB7[D31D0000]                   MOV    DH, [DSK_STATE+eDI] ; GET RATE STATE OF THIS DRIVE
6459 000023FA 80E6C0                           AND    DH,RATE_MSK         ; KEEP ONLY RATE
6460 000023FD E84E050000                       CALL   CMOS_TYPE           ; RETURN DRIVE TYPE IN AL (AL)
6461                                           ;;20/02/2015
6462                                           ;;JC   short RWV_ASSUME    ; ERROR IN CMOS
6463 00002402 7451                             jz     short RWV_ASSUME ; 20/02/2015
6464 00002404 3C01                             CMP    AL,1                ; 40 TRACK DRIVE?
6465 00002406 750D                             JNE    short RWV_1         ; NO, BYPASS CMOS VALIDITY CHECK
6466 00002408 F687[D31D0000]01                 TEST   byte [DSK_STATE+eDI], TRK_CAPA ; CHECK FOR 40 TRACK DRIVE
6467 0000240F 7413                             JZ     short RWV_2         ; YES, CMOS IS CORRECT
6468 00002411 B002                             MOV    AL,2                ; CHANGE TO 1.2M
6469 00002413 EB0F                             JMP    SHORT RWV_2
6470                                    RWV_1:
6471 00002415 720D                             JB     short RWV_2         ; NO DRIVE SPECIFIED, CONTINUE
6472 00002417 F687[D31D0000]01                 TEST    byte [DSK_STATE+eDI], TRK_CAPA ; IS IT REALLY 40 TRACK?
6473 0000241E 7504                             JNZ    short RWV_2         ; NO, 80 TRACK
6474 00002420 B001                             MOV    AL,1                ; IT IS 40 TRACK, FIX CMOS VALUE
6475 00002422 EB04                             jmp    short rwv_3
6476                                    RWV_2:
6477 00002424 08C0                             OR     AL,AL               ; TEST FOR NO DRIVE
6478 00002426 742D                             JZ     short RWV_ASSUME    ; ASSUME TYPE, USE MAX TRACK
6479                                    rwv_3:
6480 00002428 E873FEFFFF                       CALL   DR_TYPE_CHECK       ; RTN CS:BX = MEDIA/DRIVE PARAM TBL.
6481 0000242D 7226                             JC     short RWV_ASSUME    ; TYPE NOT IN TABLE (BAD CMOS)
6482
6483                                    ;-----       SEARCH FOR MEDIA/DRIVE PARAMETER TABLE
6484
6485 0000242F 57                               PUSH   eDI                 ; SAVE DRIVE #
6486 00002430 31DB                             XOR    eBX,eBX                   ; BX = INDEX TO DR_TYPE TABLE
6487 00002432 B906000000                       MOV    eCX,DR_CNT          ; CX = LOOP COUNT
```

```
6488                                    RWV_DR_SEARCH:
6489 00002437 8AA3[DB1D0000]                MOV    AH, [DR_TYPE+eBX]   ; GET DRIVE TYPE
6490 0000243D 80E47F                        AND    AH,BIT7OFF          ; MASK OUT MSB
6491 00002440 38E0                          CMP    AL,AH               ; DRIVE TYPE MATCH?
6492 00002442 750B                          JNE    short RWV_NXT_MD    ; NO, CHECK NEXT DRIVE TYPE
6493                                    RWV_DR_FND:
6494 00002444 8BBB[DC1D0000]                MOV    eDI, [DR_TYPE+eBX+1]      ; DI = MEDIA/DRIVE PARAMETER TABLE
6495                                    RWV_MD_SEARH:
6496 0000244A 3A770C                          CMP     DH, [eDI+MD.RATE]       ; MATCH?
6497 0000244D 741B                          JE     short RWV_MD_FND    ; YES, GO GET 1ST SPECIFY BYTE
6498                                    RWV_NXT_MD:
6499                                        ;ADD   BX,3                ; CHECK NEXT DRIVE TYPE
6500 0000244F 83C305                        add    eBX, 5
6501 00002452 E2E3                          LOOP   RWV_DR_SEARCH
6502 00002454 5F                            POP    eDI                 ; RESTORE DRIVE #
6503
6504                                    ;-----      ASSUME PRIMARY DRIVE IS INSTALLED AS SHIPPED
6505
6506                                    RWV_ASSUME:
6507 00002455 BB[F91D0000]                  MOV    eBX, MD_TBL1        ; POINT TO 40 TRACK 250 KBS
6508 0000245A F687[D31D0000]01              TEST   byte [DSK_STATE+eDI], TRK_CAPA ; TEST FOR 80 TRACK
6509 00002461 740A                          JZ     short RWV_MD_FND1   ; MUST BE 40 TRACK
6510 00002463 BB[131E0000]                  MOV    eBX, MD_TBL3        ; POINT TO 80 TRACK 500 KBS
6511 00002468 EB03                          JMP    short RWV_MD_FND1   ; GO SPECIFY PARAMTERS
6512
6513                                    ;-----      CS:BX POINTS TO MEDIA/DRIVE PARAMETER TABLE
6514
6515                                    RWV_MD_FND:
6516 0000246A 89FB                          MOV    eBX,eDI                   ; BX = MEDIA/DRIVE PARAMETER TABLE
6517 0000246C 5F                            POP    eDI                 ; RESTORE DRIVE #
6518
6519                                    ;-----      SEND THE SPECIFY COMMAND TO THE CONTROLLER
6520
6521                                    RWV_MD_FND1:
6522 0000246D E882FEFFFF                    CALL   SEND_SPEC_MD
6523 00002472 E864010000                    CALL   CHK_LASTRATE        ; ZF=1 ATTEMP RATE IS SAME AS LAST RATE
6524 00002477 7405                          JZ     short RWV_DBL       ; YES,SKIP SEND RATE COMMAND
6525 00002479 E83B010000                    CALL   SEND_RATE           ; SEND DATA RATE TO NEC
6526                                    RWV_DBL:
6527 0000247E 53                            PUSH   eBX                 ; SAVE MEDIA/DRIVE PARAM TBL ADDRESS
6528 0000247F E823040000                    CALL   SETUP_DBL           ; CHECK FOR DOUBLE STEP
6529 00002484 5B                            POP    eBX                 ; RESTORE ADDRESS
6530 00002485 7226                          JC     short CHK_RET       ; ERROR FROM READ ID, POSSIBLE RETRY
6531 00002487 6658                          POP    AX                  ; RESTORE NEC, DMA COMMAND
6532 00002489 6650                          PUSH   AX                  ; SAVE NEC COMMAND
6533 0000248B 53                            PUSH   eBX                 ; SAVE MEDIA/DRIVE PARAM TBL ADDRESS
6534 0000248C E861010000                    CALL   DMA_SETUP           ; SET UP THE DMA
6535 00002491 5B                            POP    eBX
6536 00002492 6658                          POP    AX                  ; RESTORE NEC COMMAND
```

```
6537 00002494 722F                      JC     short RWV_BAC       ; CHECK FOR DMA BOUNDARY ERROR
6538 00002496 6650                      PUSH   AX                  ; SAVE NEC COMMAND
6539 00002498 53                        PUSH   eBX                 ; SAVE MEDIA/DRIVE PARAM TBL ADDRESS
6540 00002499 E83D020000                CALL   NEC_INIT            ; INITIALIZE NEC
6541 0000249E 5B                        POP    eBX                 ; RESTORE ADDRESS
6542 0000249F 720C                      JC     short CHK_RET       ; ERROR - EXIT
6543 000024A1 E867020000                CALL   RWV_COM                   ; OP CODE COMMON TO READ/WRITE/VERIFY
6544 000024A6 7205                      JC     short CHK_RET       ; ERROR - EXIT
6545 000024A8 E8AC020000                CALL   NEC_TERM            ; TERMINATE, GET STATUS, ETC.
6546                                 CHK_RET:
6547 000024AD E84B030000                CALL   RETRY               ; CHECK FOR, SETUP RETRY
6548 000024B2 6658                      POP    AX                  ; RESTORE READ/WRITE/VERIFY PARAMETER
6549 000024B4 7305                      JNC    short RWV_END       ; CY = 0 NO RETRY
6550 000024B6 E928FFFFFF                  JMP     DO_AGAIN                ; CY = 1 MEANS RETRY
6551                                 RWV_END:
6552 000024BB E8F5020000                CALL   DSTATE              ; ESTABLISH STATE IF SUCCESSFUL
6553 000024C0 E888030000                CALL   NUM_TRANS           ; AL = NUMBER TRANSFERRED
6554                                 RWV_BAC:                       ; BAD DMA ERROR ENTRY
6555 000024C5 6650                      PUSH   AX                  ; SAVE NUMBER TRANSFERRED
6556 000024C7 E879FEFFFF                CALL   XLAT_OLD            ; TRANSLATE STATE TO COMPATIBLE MODE
6557 000024CC 6658                      POP    AX                  ; RESTORE NUMBER TRANSFERRED
6558 000024CE E8B2030000                CALL   SETUP_END           ; VARIOUS CLEANUPS
6559 000024D3 C3                        RETn
6560
6561                                 ;--------------------------------------------------------------------------
6562                                 ; SETUP_STATE:     INITIALIZES START AND END RATES.
6563                                 ;--------------------------------------------------------------------------
6564                                 SETUP_STATE:
6565 000024D4 F687[D31D0000]10          TEST   byte [DSK_STATE+eDI], MED_DET ; MEDIA DETERMINED ?
6566 000024DB 7537                      JNZ    short J1C           ; NO STATES IF DETERMINED
6567 000024DD 66B84000                    MOV     AX,(RATE_500*256)+RATE_300  ; AH = START RATE, AL = END RATE
6568 000024E1 F687[D31D0000]04          TEST   byte [DSK_STATE+eDI],DRV_DET ; DRIVE ?
6569 000024E8 740D                      JZ     short AX_SET        ; DO NOT KNOW DRIVE
6570 000024EA F687[D31D0000]02          TEST   byte [DSK_STATE+eDI], FMT_CAPA ; MULTI-RATE?
6571 000024F1 7504                      JNZ    short AX_SET        ; JUMP IF YES
6572 000024F3 66B88080                    MOV     AX,RATE_250*257         ; START A END RATE 250 FOR 360 DRIVE
6573                                 AX_SET:
6574 000024F7 80A7[D31D0000]1F          AND    byte [DSK_STATE+eDI], ~(RATE_MSK+DBL_STEP) ; TURN OFF THE RATE
6575 000024FE 08A7[D31D0000]            OR     [DSK_STATE+eDI], AH ; RATE FIRST TO TRY
6576 00002504 8025[D21D0000]F3          AND    byte [LASTRATE], ~STRT_MSK ; ERASE LAST TO TRY RATE BITS
6577 0000250B C0C804                    ROR    AL,4                ; TO OPERATION LAST RATE LOCATION
6578 0000250E 0805[D21D0000]            OR     [LASTRATE], AL            ; LAST RATE
6579                                 J1C:
6580 00002514 C3                        RETn
6581
6582                                 ;--------------------------------------------------------------------------
6583                                 ;  FMT_INIT: ESTABLISH STATE IF UNESTABLISHED AT FORMAT TIME.
6584                                 ;--------------------------------------------------------------------------
6585                                 FMT_INIT:
```

```
6586 00002515 F687[D31D0000]10            TEST   byte [DSK_STATE+eDI], MED_DET ; IS MEDIA ESTABLISHED
6587 0000251C 7546                        JNZ    short F1_OUT        ; IF SO RETURN
6588 0000251E E82D040000                  CALL   CMOS_TYPE           ; RETURN DRIVE TYPE IN AL
6589                                      ;; 20/02/2015
6590                                      ;;JC   short CL_DRV        ; ERROR IN CMOS ASSUME NO DRIVE
6591 00002523 7440                        jz     short CL_DRV ;; 20/02/2015
6592 00002525 FEC8                        DEC    AL                  ; MAKE ZERO ORIGIN
6593                                      ;;JS   short CL_DRV        ; NO DRIVE IF AL 0
6594 00002527 8AA7[D31D0000]              MOV    AH, [DSK_STATE+eDI] ; AH = CURRENT STATE
6595 0000252D 80E40F                      AND    AH, ~(MED_DET+DBL_STEP+RATE_MSK) ; CLEAR
6596 00002530 08C0                        OR     AL,AL               ; CHECK FOR 360
6597 00002532 7505                        JNZ    short N_360         ; IF 360 WILL BE 0
6598 00002534 80CC90                      OR     AH,MED_DET+RATE_250 ; ESTABLISH MEDIA
6599 00002537 EB25                        JMP    SHORT SKP_STATE           ; SKIP OTHER STATE PROCESSING
6600                                  N_360:
6601 00002539 FEC8                        DEC    AL                  ; 1.2 M DRIVE
6602 0000253B 7505                        JNZ    short N_12          ; JUMP IF NOT
6603                                  F1_RATE:
6604 0000253D 80CC10                      OR     AH,MED_DET+RATE_500 ; SET FORMAT RATE
6605 00002540 EB1C                        JMP    SHORT SKP_STATE           ; SKIP OTHER STATE PROCESSING
6606                                  N_12:
6607 00002542 FEC8                        DEC    AL                  ; CHECK FOR TYPE 3
6608 00002544 750F                        JNZ    short N_720         ; JUMP IF NOT
6609 00002546 F6C404                      TEST   AH,DRV_DET          ; IS DRIVE DETERMINED
6610 00002549 7410                        JZ     short ISNT_12       ; TREAT AS NON 1.2 DRIVE
6611 0000254B F6C402                      TEST   AH,FMT_CAPA         ; IS 1.2M
6612 0000254E 740B                        JZ     short ISNT_12       ; JUMP IF NOT
6613 00002550 80CC50                      OR     AH,MED_DET+RATE_300 ; RATE 300
6614 00002553 EB09                        JMP    SHORT SKP_STATE           ; CONTINUE
6615                                  N_720:
6616 00002555 FEC8                        DEC    AL                  ; CHECK FOR TYPE 4
6617 00002557 750C                        JNZ    short CL_DRV        ; NO DRIVE, CMOS BAD
6618 00002559 EBE2                        JMP    SHORT F1_RATE
6619                                  ISNT_12:
6620 0000255B 80CC90                      OR     AH,MED_DET+RATE_250 ; MUST BE RATE 250
6621
6622                                  SKP_STATE:
6623 0000255E 88A7[D31D0000]              MOV    [DSK_STATE+eDI], AH ; STORE AWAY
6624                                  F1_OUT:
6625 00002564 C3                          RETn
6626                                  CL_DRV:
6627 00002565 30E4                        XOR    AH,AH               ; CLEAR STATE
6628 00002567 EBF5                        JMP    SHORT SKP_STATE           ; SAVE IT
6629
6630                                  ;--------------------------------------------------------------------------
6631                                  ; MED_CHANGE
6632                                  ;     CHECKS FOR MEDIA CHANGE, RESETS MEDIA CHANGE,
6633                                  ;     CHECKS MEDIA CHANGE AGAIN.
6634                                  ;
```

```
                                ; ON EXIT:  CY = 1 MEANS MEDIA CHANGE OR TIMEOUT
                                ;           @DSKETTE_STATUS = ERROR CODE
                                ;--------------------------------------------------------------------------------
                                MED_CHANGE:
00002569 E87E060000                  CALL   READ_DSKCHNG        ; READ DISK CHANCE LINE STATE
0000256E 7447                        JZ     short MC_OUT        ; BYPASS HANDLING DISK CHANGE LINE
00002570 80A7[D31D0000]EF            AND    byte [DSK_STATE+eDI], ~MED_DET ; CLEAR STATE FOR THIS DRIVE

                                ;     THIS SEQUENCE ENSURES WHENEVER A DISKETTE IS CHANGED THAT
                                ;     ON THE NEXT OPERATION THE REQUIRED MOTOR START UP TIME WILL
                                ;     BE WAITED. (DRIVE MOTOR MAY GO OFF UPON DOOR OPENING).

00002577 6689F9                      MOV    CX,DI               ; CL = DRIVE 0
0000257A B001                        MOV    AL,1                ; MOTOR ON BIT MASK
0000257C D2E0                        SHL    AL,CL               ; TO APPROPRIATE POSITION
0000257E F6D0                        NOT    AL                  ; KEEP ALL BUT MOTOR ON
00002580 FA                          CLI                        ; NO INTERRUPTS
00002581 2005[C31D0000]              AND    [MOTOR_STATUS], AL  ; TURN MOTOR OFF INDICATOR
00002587 FB                          STI                        ; INTERRUPTS ENABLED
00002588 E811040000                  CALL   MOTOR_ON            ; TURN MOTOR ON

                                ;-----      THIS SEQUENCE OF SEEKS IS USED TO RESET DISKETTE CHANGE SIGNAL

0000258D E88DF9FFFF                  CALL   DSK_RESET           ; RESET NEC
00002592 B501                        MOV    CH,01H              ; MOVE TO CYLINDER 1
00002594 E8F5040000                  CALL   SEEK                ; ISSUE SEEK
00002599 30ED                        XOR    CH,CH               ; MOVE TO CYLINDER 0
0000259B E8EE040000                  CALL   SEEK                ; ISSUE SEEK
000025A0 C605[C51D0000]06            MOV    byte [DSKETTE_STATUS], MEDIA_CHANGE ; STORE IN STATUS
                                OK1:
000025A7 E840060000                  CALL   READ_DSKCHNG        ; CHECK MEDIA CHANGED AGAIN
000025AC 7407                        JZ     short OK2           ; IF ACTIVE, NO DISKETTE, TIMEOUT
                                OK4:
000025AE C605[C51D0000]80            MOV    byte [DSKETTE_STATUS], TIME_OUT ; TIMEOUT IF DRIVE EMPTY
                                OK2:
000025B5 F9                          STC                        ; MEDIA CHANGED, SET CY
000025B6 C3                          RETn
                                MC_OUT:
000025B7 F8                          CLC                        ; NO MEDIA CHANGED, CLEAR CY
000025B8 C3                          RETn

                                ;--------------------------------------------------------------------------------
                                ; SEND_RATE
                                ;     SENDS DATA RATE COMMAND TO NEC
                                ; ON ENTRY:  DI = DRIVE #
                                ; ON EXIT:   NONE
                                ; REGISTERS ALTERED: DX
                                ;--------------------------------------------------------------------------------
                                SEND_RATE:
```

```
6684 000025B9 6650                      PUSH   AX                  ; SAVE REG.
6685 000025BB 8025[D21D0000]3F          AND    byte [LASTRATE], ~SEND_MSK ; ELSE CLEAR LAST RATE ATTEMPTED
6686 000025C2 8A87[D31D0000]            MOV    AL, [DSK_STATE+eDI] ; GET RATE STATE OF THIS DRIVE
6687 000025C8 24C0                      AND    AL,SEND_MSK         ; KEEP ONLY RATE BITS
6688 000025CA 0805[D21D0000]            OR     [LASTRATE], AL            ; SAVE NEW RATE FOR NEXT CHECK
6689 000025D0 C0C002                    ROL    AL,2                ; MOVE TO BIT OUTPUT POSITIONS
6690 000025D3 66BAF703                  MOV    DX,03F7H            ; OUTPUT NEW DATA RATE
6691 000025D7 EE                        OUT    DX,AL
6692 000025D8 6658                      POP    AX                  ; RESTORE REG.
6693 000025DA C3                        RETn
6694
6695                                ;--------------------------------------------------------------------------
6696                                ; CHK_LASTRATE
6697                                ;     CHECK PREVIOUS DATE RATE SNT TO THE CONTROLLER.
6698                                ; ON ENTRY:
6699                                ;     DI = DRIVE #
6700                                ; ON EXIT:
6701                                ;     ZF =  1 DATA RATE IS THE SAME AS THE LAST RATE SENT TO NEC
6702                                ;     ZF =  0 DATA RATE IS DIFFERENT FROM LAST RATE
6703                                ; REGISTERS ALTERED: DX
6704                                ;--------------------------------------------------------------------------
6705                                CHK_LASTRATE:
6706 000025DB 6650                      PUSH   AX                  ; SAVE REG
6707 000025DD 2225[D21D0000]            AND    AH, [LASTRATE]            ; GET LAST DATA RATE SELECTED
6708 000025E3 8A87[D31D0000]            MOV    AL, [DSK_STATE+eDI] ; GET RATE STATE OF THIS DRIVE
6709 000025E9 6625C0C0                    AND     AX, SEND_MSK*257        ; KEEP ONLY RATE BITS OF BOTH
6710 000025ED 38E0                      CMP    AL, AH              ; COMPARE TO PREVIOUSLY TRIED
6711                                                               ; ZF = 1 RATE IS THE SAME
6712 000025EF 6658                      POP    AX                  ; RESTORE REG.
6713 000025F1 C3                        RETn
6714
6715                                ;--------------------------------------------------------------------------
6716                                ; DMA_SETUP
6717                                ;     THIS ROUTINE SETS UP THE DMA FOR READ/WRITE/VERIFY OPERATIONS.
6718                                ;
6719                                ; ON ENTRY:  AL = DMA COMMAND
6720                                ;
6721                                ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
6722                                ;--------------------------------------------------------------------------
6723
6724                                ; SI = Head #, # of Sectors or DASD Type
6725
6726                                ; 08/02/2015 - Protected Mode Modification
6727                                ; 06/02/2015 - 07/02/2015
6728                                ; NOTE: Buffer address must be in 1st 16MB of Physical Memory (24 bit limit).
6729                                ; (DMA Addres = Physical Address)
6730                                ; (Retro UNIX 386 v1 Kernel/System Mode Virtual Address = Physical Address)
6731                                ;
6732
```

```
                                ; 20/02/2015 modification (source: AWARD BIOS 1999, DMA_SETUP)
                                ; 16/12/2014 (IODELAY)

                                DMA_SETUP:
                                ;; 20/02/2015
000025F2 8B5504                       mov   edx, [ebp+4]        ; Buffer address
000025F5 F7C20000F0FF                 test  edx, 0FFF00000h           ; 16 MB limit
000025FB 756F                         jnz   short dma_bnd_err_stc
                                      ;
000025FD 6650                         push  ax                  ; DMA command
000025FF 6652                         push  dx                  ; *
00002601 B203                         mov   dl, 3               ; GET BYTES/SECTOR PARAMETER
00002603 E851030000                   call  GET_PARM            ;
00002608 88E1                         mov   cl, ah                    ; SHIFT COUNT (0=128, 1=256, 2=512 ETC)
0000260A 6689F0                       mov   ax, si              ; Sector count
0000260D 88C4                         mov   ah, al              ; AH =  # OF SECTORS
0000260F 28C0                         sub   al, al              ; AL = 0, AX = # SECTORS * 256
00002611 66D1E8                       shr   ax, 1               ; AX = # SECTORS * 128
00002614 66D3E0                       shl   ax, cl              ; SHIFT BY PARAMETER VALUE
00002617 6648                         dec   ax                  ; -1 FOR DMA VALUE
00002619 6689C1                       mov   cx, ax
0000261C 665A                         pop   dx                  ; *
0000261E 6658                         pop   ax
00002620 3C42                         cmp   al, 42h
00002622 7507                           jne     short NOT_VERF
00002624 BA0000FF00                   mov   edx, 0FF0000h
00002629 EB08                         jmp   short J33
                                NOT_VERF:
0000262B 6601CA                       add   dx, cx              ; check for overflow
0000262E 723D                         jc    short dma_bnd_err
                                      ;
00002630 6629CA                       sub   dx, cx              ; Restore start address
                                J33:
00002633 FA                           CLI                       ; DISABLE INTERRUPTS DURING DMA SET-UP
00002634 E60C                         OUT   DMA+12,AL           ; SET THE FIRST/LA5T F/F
                                      IODELAY                         ; WAIT FOR I/O
00002636 EB00                 <1>  jmp short $+2
00002638 EB00                 <1>  jmp short $+2
0000263A E60B                         OUT   DMA+11,AL           ; OUTPUT THE MODE BYTE
0000263C 89D0                         mov   eax, edx            ; Buffer address
0000263E E604                         OUT   DMA+4,AL            ; OUTPUT LOW ADDRESS
                                      IODELAY                         ; WAIT FOR I/O
00002640 EB00                 <1>  jmp short $+2
00002642 EB00                 <1>  jmp short $+2
00002644 88E0                         MOV   AL,AH
00002646 E604                         OUT   DMA+4,AL            ; OUTPUT HIGH ADDRESS
00002648 C1E810                       shr   eax, 16
                                      IODELAY                         ; I/O WAIT STATE
0000264B EB00                 <1>  jmp short $+2
```

```
6782 0000264D EB00              <1>  jmp short $+2
6783 0000264F E681                   OUT   081H,AL                  ; OUTPUT highest BITS TO PAGE REGISTER
6784                                 IODELAY
6785 00002651 EB00              <1>  jmp short $+2
6786 00002653 EB00              <1>  jmp short $+2
6787 00002655 6689C8                 mov   ax, cx               ; Byte count - 1
6788 00002658 E605                   OUT   DMA+5,AL             ; LOW BYTE OF COUNT
6789                                 IODELAY                         ; WAIT FOR I/O
6790 0000265A EB00              <1>  jmp short $+2
6791 0000265C EB00              <1>  jmp short $+2
6792 0000265E 88E0                   MOV   AL, AH
6793 00002660 E605                   OUT   DMA+5,AL             ; HIGH BYTE OF COUNT
6794                                 IODELAY
6795 00002662 EB00              <1>  jmp short $+2
6796 00002664 EB00              <1>  jmp short $+2
6797 00002666 FB                     STI                        ; RE-ENABLE INTERRUPTS
6798 00002667 B002                   MOV   AL, 2                ; MODE FOR 8237
6799 00002669 E60A                   OUT   DMA+10, AL           ; INITIALIZE THE DISKETTE CHANNEL
6800 0000266B C3                     retn
6801                            dma_bnd_err_stc:
6802 0000266C F9                     stc
6803                            dma_bnd_err:
6804 0000266D C605[C51D0000]09       MOV   byte [DSKETTE_STATUS], DMA_BOUNDARY ; SET ERROR
6805 00002674 C3                     RETn                       ; CY SET BY ABOVE IF ERROR
6806
6807                            ;; 16/12/2014
6808                            ;;   CLI                        ; DISABLE INTERRUPTS DURING DMA SET-UP
6809                            ;;   OUT   DMA+12,AL            ; SET THE FIRST/LA5T F/F
6810                            ;;   ;JMP  $+2                  ; WAIT FOR I/O
6811                            ;;   IODELAY
6812                            ;;   OUT   DMA+11,AL            ; OUTPUT THE MODE BYTE
6813                            ;;   ;SIODELAY
6814                            ;;     ;CMP AL, 42H                  ; DMA VERIFY COMMAND
6815                            ;;     ;JNE short NOT_VERF           ; NO
6816                            ;;     ;XOR AX, AX              ; START ADDRESS
6817                            ;;     ;JMP SHORT J33
6818                            ;;;NOT_VERF:
6819                            ;;   ;MOV  AX,ES                ; GET THE ES VALUE
6820                            ;;   ;ROL  AX,4                 ; ROTATE LEFT
6821                            ;;   ;MOV  CH,AL                ; GET HIGHEST NIBBLE OF ES TO CH
6822                            ;;   ;AND  AL,11110000B         ; ZERO THE LOW NIBBLE FROM SEGMENT
6823                            ;;   ;ADD  AX,[BP+2]            ; TEST FOR CARRY FROM ADDITION
6824                            ;;   mov   eax, [ebp+4] ; 06/02/2015
6825                            ;;   ;JNC  short J33
6826                            ;;   ;INC  CH                   ; CARRY MEANS HIGH 4 BITS MUST BE INC
6827                            ;;;J33:
6828                            ;;   PUSH  eAX                  ; SAVE START ADDRESS
6829                            ;;   OUT   DMA+4,AL             ; OUTPUT LOW ADDRESS
6830                            ;;   ;JMP  $+2                  ; WAIT FOR I/O
```

```
;;    IODELAY
;;    MOV   AL,AH
;;    OUT   DMA+4,AL           ; OUTPUT HIGH ADDRESS
;;    shr   eax, 16          ; 07/02/2015
;;    ;MOV  AL,CH              ; GET HIGH 4 BITS
;;    ;JMP  $+2                ; I/O WAIT STATE
;;    IODELAY
;;    ;AND  AL,00001111B
;;    OUT   081H,AL                  ; OUTPUT HIGH 4 BITS TO PAGE REGISTER
;;    ;SIODELAY
;;
;;;----- DETERMINE COUNT
;;    sub   eax, eax ; 08/02/2015
;;    MOV   AX, SI             ; AL =  # OF SECTORS
;;    XCHG  AL, AH             ; AH =  # OF SECTORS
;;    SUB   AL, AL             ; AL = 0, AX = # SECTORS * 256
;;    SHR   AX, 1              ; AX = # SECTORS * 128
;;    PUSH  AX                 ; SAVE # OF SECTORS * 128
;;    MOV   DL, 3              ; GET BYTES/SECTOR PARAMETER
;;    CALL  GET_PARM           ; "
;;    MOV   CL,AH              ; SHIFT COUNT (0=128, 1=256, 2=512 ETC)
;;    POP   AX                 ; AX = # SECTORS * 128
;;    SHL   AX,CL              ; SHIFT BY PARAMETER VALUE
;;    DEC   AX                 ; -1 FOR DMA VALUE
;;    PUSH  eAX  ; 08/02/2015  ; SAVE COUNT VALUE
;;    OUT   DMA+5,AL           ; LOW BYTE OF COUNT
;;    ;JMP  $+2                ; WAIT FOR I/O
;;    IODELAY
;;    MOV   AL, AH
;;    OUT   DMA+5,AL           ; HIGH BYTE OF COUNT
;;    ;IODELAY
;;    STI                      ; RE-ENABLE INTERRUPTS
;;    POP   eCX  ; 08/02/2015  ; RECOVER COUNT VALUE
;;    POP   eAX  ; 08/02/2015  ; RECOVER ADDRESS VALUE
;;    ;ADD  AX, CX             ; ADD, TEST FOR 64K OVERFLOW
;;    add   ecx, eax ; 08/02/2015
;;    MOV   AL, 2              ; MODE FOR 8237
;;    ;JMP  $+2                ; WAIT FOR I/O
;;    SIODELAY
;;    OUT   DMA+10, AL         ; INITIALIZE THE DISKETTE CHANNEL
;;    ;JNC  short NO_BAD       ; CHECK FOR ERROR
;;    jc    short dma_bnd_err ; 08/02/2015
;;    and   ecx, 0FFF00000h ; 16 MB limit
;;    jz    short NO_BAD
;;dma_bnd_err:
;;    MOV   byte [DSKETTE_STATUS], DMA_BOUNDARY ; SET ERROR
;;NO_BAD:
;;    RETn                     ; CY SET BY ABOVE IF ERROR
```

```
                                ;--------------------------------------------------------------------------
                                ; FMTDMA_SET
                                ;     THIS ROUTINE SETS UP THE DMA CONTROLLER FOR A FORMAT OPERATION.
                                ;
                                ; ON ENTRY:  NOTHING REQUIRED
                                ;
                                ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
                                ;--------------------------------------------------------------------------

                                FMTDMA_SET:
                                ;; 20/02/2015 modification
00002675 8B5504                       mov    edx, [ebp+4]        ; Buffer address
00002678 F7C20000F0FF                 test   edx, 0FFF00000h           ; 16 MB limit
0000267E 75EC                         jnz    short dma_bnd_err_stc
                                      ;
00002680 6652                         push   dx                  ; *
00002682 B204                         mov    DL, 4               ; SECTORS/TRACK VALUE IN PARM TABLE
00002684 E8D0020000                   call   GET_PARM            ; "
00002689 88E0                         mov    al, ah              ; AL = SECTORS/TRACK VALUE
0000268B 28E4                         sub    ah, ah              ; AX = SECTORS/TRACK VALUE
0000268D 66C1E002                     shl    ax, 2               ; AX = SEC/TRK * 4 (OFFSET C,H,R,N)
00002691 6648                         dec    ax                  ; -1 FOR DMA VALUE
00002693 6689C1                       mov    cx, ax
00002696 665A                         pop    dx                  ; *
00002698 6601CA                       add    dx, cx              ; check for overflow
0000269B 72D0                         jc     short dma_bnd_err
                                      ;
0000269D 6629CA                       sub    dx, cx              ; Restore start address
                                      ;
000026A0 B04A                         MOV    AL, 04AH            ; WILL WRITE TO THE DISKETTE
000026A2 FA                           CLI                        ; DISABLE INTERRUPTS DURING DMA SET-UP
000026A3 E60C                         OUT    DMA+12,AL           ; SET THE FIRST/LA5T F/F
                                      IODELAY                          ; WAIT FOR I/O
000026A5 EB00                 <1>  jmp short $+2
000026A7 EB00                 <1>  jmp short $+2
000026A9 E60B                         OUT    DMA+11,AL           ; OUTPUT THE MODE BYTE
000026AB 89D0                         mov    eax, edx            ; Buffer address
000026AD E604                         OUT    DMA+4,AL            ; OUTPUT LOW ADDRESS
                                      IODELAY                          ; WAIT FOR I/O
000026AF EB00                 <1>  jmp short $+2
000026B1 EB00                 <1>  jmp short $+2
000026B3 88E0                         MOV    AL,AH
000026B5 E604                         OUT    DMA+4,AL            ; OUTPUT HIGH ADDRESS
000026B7 C1E810                       shr    eax, 16
                                      IODELAY                          ; I/O WAIT STATE
000026BA EB00                 <1>  jmp short $+2
000026BC EB00                 <1>  jmp short $+2
000026BE E681                         OUT    081H,AL                  ; OUTPUT highest BITS TO PAGE REGISTER
                                      IODELAY
```

```
6929 000026C0 EB00              <1>  jmp short $+2
6930 000026C2 EB00              <1>  jmp short $+2
6931 000026C4 6689C8                      mov   ax, cx              ; Byte count - 1
6932 000026C7 E605                        OUT   DMA+5,AL            ; LOW BYTE OF COUNT
6933                                      IODELAY                         ; WAIT FOR I/O
6934 000026C9 EB00              <1>  jmp short $+2
6935 000026CB EB00              <1>  jmp short $+2
6936 000026CD 88E0                        MOV   AL, AH
6937 000026CF E605                        OUT   DMA+5,AL            ; HIGH BYTE OF COUNT
6938                                      IODELAY
6939 000026D1 EB00              <1>  jmp short $+2
6940 000026D3 EB00              <1>  jmp short $+2
6941 000026D5 FB                          STI                       ; RE-ENABLE INTERRUPTS
6942 000026D6 B002                        MOV   AL, 2               ; MODE FOR 8237
6943 000026D8 E60A                        OUT   DMA+10, AL          ; INITIALIZE THE DISKETTE CHANNEL
6944 000026DA C3                          retn
6945
6946                                  ;; 08/02/2015 - Protected Mode Modification
6947                                  ;;    MOV   AL, 04AH            ; WILL WRITE TO THE DISKETTE
6948                                  ;;    CLI                       ; DISABLE INTERRUPTS DURING DMA SET-UP
6949                                  ;;    OUT   DMA+12,AL           ; SET THE FIRST/LA5T F/F
6950                                  ;;    ;JMP  $+2                 ; WAIT FOR I/O
6951                                  ;;    IODELAY
6952                                  ;;    OUT   DMA+11,AL           ; OUTPUT THE MODE BYTE
6953                                  ;;    ;MOV  AX,ES               ; GET THE ES VALUE
6954                                  ;;    ;ROL  AX,4                ; ROTATE LEFT
6955                                  ;;    ;MOV  CH,AL               ; GET HIGHEST NIBBLE OF ES TO CH
6956                                  ;;    ;AND  AL,11110000B        ; ZERO THE LOW NIBBLE FROM SEGMENT
6957                                  ;;    ;ADD  AX,[BP+2]           ; TEST FOR CARRY FROM ADDITION
6958                                  ;;    ;JNC  short J33A
6959                                  ;;    ;INC  CH                  ; CARRY MEANS HIGH 4 BITS MUST BE INC
6960                                  ;;    mov   eax, [ebp+4] ; 08/02/2015
6961                                  ;;;J33A:
6962                                  ;;    PUSH  eAX ; 08/02/2015    ; SAVE START ADDRESS
6963                                  ;;    OUT   DMA+4,AL            ; OUTPUT LOW ADDRESS
6964                                  ;;    ;JMP  $+2                 ; WAIT FOR I/O
6965                                  ;;    IODELAY
6966                                  ;;    MOV   AL,AH
6967                                  ;;    OUT   DMA+4,AL            ; OUTPUT HIGH ADDRESS
6968                                  ;;    shr   eax, 16 ; 08/02/2015
6969                                  ;;    ;MOV  AL,CH               ; GET HIGH 4 BITS
6970                                  ;;    ;JMP  $+2                 ; I/O WAIT STATE
6971                                  ;;    IODELAY
6972                                  ;;    ;AND  AL,00001111B
6973                                  ;;    OUT   081H,AL                   ; OUTPUT HIGH 4 BITS TO PAGE REGISTER
6974                                  ;;
6975                                  ;;;----- DETERMINE COUNT
6976                                  ;;    sub   eax, eax ; 08/02/2015
6977                                  ;;    MOV   DL, 4               ; SECTORS/TRACK VALUE IN PARM TABLE
```

```
                        ;;    CALL   GET_PARM           ; "
                        ;;    XCHG   AL, AH             ; AL = SECTORS/TRACK VALUE
                        ;;    SUB    AH, AH             ; AX = SECTORS/TRACK VALUE
                        ;;    SHL    AX, 2              ; AX = SEC/TRK * 4 (OFFSET C,H,R,N)
                        ;;    DEC    AX                 ; -1 FOR DMA VALUE
                        ;;    PUSH   eAX    ; 08/02/2015 ; SAVE # OF BYTES TO BE TRANSFERED
                        ;;    OUT    DMA+5,AL           ; LOW BYTE OF COUNT
                        ;;    ;JMP   $+2                ; WAIT FOR I/O
                        ;;    IODELAY
                        ;;    MOV    AL, AH
                        ;;    OUT    DMA+5,AL           ; HIGH BYTE OF COUNT
                        ;;    STI                       ; RE-ENABLE INTERRUPTS
                        ;;    POP    eCX    ; 08/02/2015 ; RECOVER COUNT VALUE
                        ;;    POP    eAX    ; 08/02/2015 ; RECOVER ADDRESS VALUE
                        ;;    ;ADD   AX, CX             ; ADD, TEST FOR 64K OVERFLOW
                        ;;    add    ecx, eax ; 08/02/2015
                        ;;    MOV    AL, 2              ; MODE FOR 8237
                        ;;    ;JMP   $+2                ; WAIT FOR I/O
                        ;;    SIODELAY
                        ;;    OUT    DMA+10, AL         ; INITIALIZE THE DISKETTE CHANNEL
                        ;;    ;JNC   short FMTDMA_OK          ; CHECK FOR ERROR
                        ;;    jc     short fmtdma_bnd_err ; 08/02/2015
                        ;;    and    ecx, 0FFF00000h  ; 16 MB limit
                        ;;    jz     short FMTDMA_OK
                        ;;    stc    ; 20/02/2015
                        ;;fmtdma_bnd_err:
                        ;;    MOV    byte [DSKETTE_STATUS], DMA_BOUNDARY ; SET ERROR
                        ;;FMTDMA_OK:
                        ;;    RETn                      ; CY SET BY ABOVE IF ERROR

                        ;--------------------------------------------------------------------------------
                        ; NEC_INIT
                        ;     THIS ROUTINE SEEKS TO THE REQUESTED TRACK AND INITIALIZES
                        ;     THE NEC FOR THE READ/WRITE/VERIFY/FORMAT OPERATION.
                        ;
                        ; ON ENTRY:  AH = NEC COMMAND TO BE PERFORMED
                        ;
                        ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
                        ;--------------------------------------------------------------------------------
                        NEC_INIT:
000026DB 6650                 PUSH   AX                 ; SAVE NEC COMMAND
000026DD E8BC020000           CALL   MOTOR_ON           ; TURN MOTOR ON FOR SPECIFIC DRIVE

                        ;-----      DO THE SEEK OPERATION

000026E2 8A6D01               MOV    CH,[eBP+1]         ; CH = TRACK #
000026E5 E8A4030000           CALL   SEEK               ; MOVE TO CORRECT TRACK
000026EA 6658                 POP    AX                 ; RECOVER COMMAND
000026EC 721E                 JC     short ER_1         ; ERROR ON SEEK
```

```
7027 000026EE BB[0C270000]              MOV    eBX, ER_1           ; LOAD ERROR ADDRESS
7028 000026F3 53                        PUSH   eBX                 ; PUSH NEC_OUT ERROR RETURN
7029
7030                            ;-----      SEND OUT THE PARAMETERS TO THE CONTROLLER
7031
7032 000026F4 E866030000                CALL   NEC_OUTPUT          ; OUTPUT THE OPERATION COMMAND
7033 000026F9 6689F0                    MOV    AX,SI               ; AH = HEAD #
7034 000026FC 89FB                      MOV    eBX,eDI                  ; BL = DRIVE #
7035 000026FE C0E402                    SAL    AH,2                ; MOVE IT TO BIT 2
7036 00002701 80E404                    AND    AH,00000100B        ; ISOLATE THAT BIT
7037 00002704 08DC                      OR     AH,BL               ; OR IN THE DRIVE NUMBER
7038 00002706 E854030000                CALL   NEC_OUTPUT          ; FALL THRU CY SET IF ERROR
7039 0000270B 5B                        POP    eBX                 ; THROW AWAY ERROR RETURN
7040                            ER_1:
7041 0000270C C3                        RETn
7042
7043                            ;--------------------------------------------------------------------------
7044                            ; RWV_COM
7045                            ;     THIS ROUTINE SENDS PARAMETERS TO THE NEC SPECIFIC TO THE
7046                            ;     READ/WRITE/VERIFY OPERATIONS.
7047                            ;
7048                            ; ON ENTRY:  CS:BX = ADDRESS OF MEDIA/DRIVE PARAMETER TABLE
7049                            ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
7050                            ;--------------------------------------------------------------------------
7051                            RWV_COM:
7052 0000270D B8[58270000]              MOV    eAX, ER_2           ; LOAD ERROR ADDRESS
7053 00002712 50                        PUSH   eAX                 ; PUSH NEC_OUT ERROR RETURN
7054 00002713 8A6501                    MOV    AH,[eBP+1]          ; OUTPUT TRACK #
7055 00002716 E844030000                CALL   NEC_OUTPUT
7056 0000271B 6689F0                    MOV    AX,SI               ; OUTPUT HEAD #
7057 0000271E E83C030000                CALL   NEC_OUTPUT
7058 00002723 8A6500                      MOV     AH,[eBP]                ; OUTPUT SECTOR #
7059 00002726 E834030000                CALL   NEC_OUTPUT
7060 0000272B B203                      MOV    DL,3                ; BYTES/SECTOR PARAMETER FROM BLOCK
7061 0000272D E827020000                CALL   GET_PARM            ; ... TO THE NEC
7062 00002732 E828030000                CALL   NEC_OUTPUT          ; OUTPUT TO CONTROLLER
7063 00002737 B204                      MOV    DL,4                ; EOT PARAMETER FROM BLOCK
7064 00002739 E81B020000                CALL   GET_PARM            ; ... TO THE NEC
7065 0000273E E81C030000                CALL   NEC_OUTPUT          ; OUTPUT TO CONTROLLER
7066 00002743 8A6305                      MOV     AH, [eBX+MD.GAP]        ; GET GAP LENGTH
7067                            _R15:
7068 00002746 E814030000                CALL   NEC_OUTPUT
7069 0000274B B206                      MOV    DL,6                ; DTL PARAMETER PROM BLOCK
7070 0000274D E807020000                CALL   GET_PARM            ;  TO THE NEC
7071 00002752 E808030000                CALL   NEC_OUTPUT          ; OUTPUT TO CONTROLLER
7072 00002757 58                        POP    eAX                 ; THROW AWAY ERROR EXIT
7073                            ER_2:
7074 00002758 C3                        RETn
7075
```

```
7076                                    ;--------------------------------------------------------------------------------
7077                                    ; NEC_TERM
7078                                    ;     THIS ROUTINE WAITS FOR THE OPERATION THEN ACCEPTS THE STATUS
7079                                    ;     FROM THE NEC FOR THE READ/WRITE/VERIFY/FORWAT OPERATION.
7080                                    ;
7081                                    ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
7082                                    ;--------------------------------------------------------------------------------
7083                                    NEC_TERM:
7084
7085                                    ;-----      LET THE OPERATION HAPPEN
7086
7087 00002759 56                              PUSH   eSI                 ; SAVE HEAD #, # OF SECTORS
7088 0000275A E802040000                      CALL   WAIT_INT            ; WAIT FOR THE INTERRUPT
7089 0000275F 9C                              PUSHF
7090 00002760 E829040000                      CALL   RESULTS                   ; GET THE NEC STATUS
7091 00002765 724B                            JC     short SET_END_POP
7092 00002767 9D                              POPF
7093 00002768 723E                            JC     short SET_END       ; LOOK FOR ERROR
7094
7095                                    ;-----      CHECK THE RESULTS RETURNED BY THE CONTROLLER
7096
7097 0000276A FC                              CLD                        ; SET THE CORRECT DIRECTION
7098 0000276B BE[C61D0000]                    MOV    eSI, NEC_STATUS           ; POINT TO STATUS FIELD
7099 00002770 AC                              lodsb                      ; GET ST0
7100 00002771 24C0                            AND    AL,11000000B        ; TEST FOR NORMAL TERMINATION
7101 00002773 7433                            JZ     short SET_END
7102 00002775 3C40                            CMP    AL,01000000B        ; TEST FOR ABNORMAL TERMINATION
7103 00002777 7527                            JNZ    short J18           ; NOT ABNORMAL, BAD NEC
7104
7105                                    ;-----      ABNORMAL TERMINATION, FIND OUT WHY
7106
7107 00002779 AC                              lodsb                      ; GET ST1
7108 0000277A D0E0                            SAL    AL,1                ; TEST FOR EDT FOUND
7109 0000277C B404                            MOV    AH,RECORD_NOT_FND
7110 0000277E 7222                            JC     short J19
7111 00002780 C0E002                          SAL    AL,2
7112 00002783 B410                            MOV    AH,BAD_CRC
7113 00002785 721B                            JC     short J19
7114 00002787 D0E0                            SAL    AL,1                ; TEST FOR DMA OVERRUN
7115 00002789 B408                            MOV    AH,BAD_DMA
7116 0000278B 7215                            JC     short J19
7117 0000278D C0E002                          SAL    AL,2                ; TEST FOR RECORD NOT FOUND
7118 00002790 B404                            MOV    AH,RECORD_NOT_FND
7119 00002792 720E                            JC     short J19
7120 00002794 D0E0                            SAL    AL,1
7121 00002796 B403                            MOV    AH,WRITE_PROTECT    ; TEST FOR WRITE_PROTECT
7122 00002798 7208                            JC     short J19
7123 0000279A D0E0                            SAL    AL,1                ; TEST MISSING ADDRESS MARK
7124 0000279C B402                            MOV    AH,BAD_ADDR_MARK
```

```
7125 0000279E 7202                      JC      short J19
7126
7127                                    ;-----  NEC MUST HAVE FAILED
7128                                    J18:
7129 000027A0 B420                      MOV     AH,BAD_NEC
7130                                    J19:
7131 000027A2 0825[C51D0000]            OR      [DSKETTE_STATUS], AH
7132                                    SET_END:
7133 000027A8 803D[C51D0000]01          CMP     byte [DSKETTE_STATUS], 1 ; SET ERROR CONDITION
7134 000027AF F5                        CMC
7135 000027B0 5E                        POP     eSI
7136 000027B1 C3                        RETn                    ; RESTORE HEAD #, # OF SECTORS
7137
7138                                    SET_END_POP:
7139 000027B2 9D                        POPF
7140 000027B3 EBF3                      JMP     SHORT SET_END
7141
7142                                    ;--------------------------------------------------------------------------
7143                                    ; DSTATE:   ESTABLISH STATE UPON SUCCESSFUL OPERATION.
7144                                    ;--------------------------------------------------------------------------
7145                                    DSTATE:
7146 000027B5 803D[C51D0000]00          CMP     byte [DSKETTE_STATUS],0   ; CHECK FOR ERROR
7147 000027BC 753E                      JNZ     short SETBAC            ; IF ERROR JUMP
7148 000027BE 808F[D31D0000]10          OR      byte [DSK_STATE+eDI],MED_DET ; NO ERROR, MARK MEDIA AS DETERMINED
7149 000027C5 F687[D31D0000]04          TEST    byte [DSK_STATE+eDI],DRV_DET ; DRIVE DETERMINED ?
7150 000027CC 752E                      JNZ     short SETBAC        ; IF DETERMINED NO TRY TO DETERMINE
7151 000027CE 8A87[D31D0000]            MOV     AL,[DSK_STATE+eDI]  ; LOAD STATE
7152 000027D4 24C0                      AND     AL,RATE_MSK         ; KEEP ONLY RATE
7153 000027D6 3C80                      CMP     AL,RATE_250         ; RATE 250 ?
7154 000027D8 751B                      JNE     short M_12          ; NO, MUST BE 1.2M OR 1.44M DRIVE
7155
7156                                    ;-----  CHECK IF IT IS 1.44M
7157
7158 000027DA E871010000                CALL    CMOS_TYPE           ; RETURN DRIVE TYPE IN (AL)
7159                                    ;;20/02/2015
7160                                    ;;JC    short M_12          ; CMOS BAD
7161 000027DF 7414                      jz      short M_12 ;; 20/02/2015
7162 000027E1 3C04                      CMP     AL, 4               ; 1.44MB DRIVE ?
7163 000027E3 7410                      JE      short M_12          ; YES
7164                                    M_720:
7165 000027E5 80A7[D31D0000]FD          AND     byte [DSK_STATE+eDI], ~FMT_CAPA ; TURN OFF FORMAT CAPABILITY
7166 000027EC 808F[D31D0000]04          OR      byte [DSK_STATE+eDI],DRV_DET  ; MARK DRIVE DETERMINED
7167 000027F3 EB07                      JMP     SHORT SETBAC        ; BACK
7168                                    M_12:
7169 000027F5 808F[D31D0000]06          OR      byte [DSK_STATE+eDI],DRV_DET+FMT_CAPA
7170                                                                ; TURN ON DETERMINED & FMT CAPA
7171                                    SETBAC:
7172 000027FC C3                        RETn
7173
```

```
7174                                ;--------------------------------------------------------------------------
7175                                ; RETRY
7176                                ;     DETERMINES WHETHER A RETRY IS NECESSARY.
7177                                ;     IF RETRY IS REQUIRED THEN STATE INFORMATION IS UPDATED FOR RETRY.
7178                                ;
7179                                ; ON EXIT:  CY = 1 FOR RETRY, CY = 0 FOR NO RETRY
7180                                ;--------------------------------------------------------------------------
7181                                RETRY:
7182 000027FD 803D[C51D0000]00             CMP    byte [DSKETTE_STATUS],0   ; GET STATUS OF OPERATION
7183 00002804 7445                         JZ     short NO_RETRY            ; SUCCESSFUL OPERATION
7184 00002806 803D[C51D0000]80             CMP    byte [DSKETTE_STATUS],TIME_OUT ; IF TIME OUT NO RETRY
7185 0000280D 743C                         JZ     short NO_RETRY
7186 0000280F 8AA7[D31D0000]               MOV    AH,[DSK_STATE+eDI]  ; GET MEDIA STATE OF DRIVE
7187 00002815 F6C410                       TEST   AH,MED_DET          ; ESTABLISHED/DETERMINED ?
7188 00002818 7531                         JNZ    short NO_RETRY            ; IF ESTABLISHED STATE THEN TRUE ERROR
7189 0000281A 80E4C0                       AND    AH,RATE_MSK         ; ISOLATE RATE
7190 0000281D 8A2D[D21D0000]               MOV    CH,[LASTRATE]       ; GET START OPERATION STATE
7191 00002823 C0C504                       ROL    CH,4                ; TO CORRESPONDING BITS
7192 00002826 80E5C0                       AND    CH,RATE_MSK         ; ISOLATE RATE BITS
7193 00002829 38E5                         CMP    CH,AH               ; ALL RATES TRIED
7194 0000282B 741E                         JE     short NO_RETRY            ; IF YES, THEN TRUE ERROR
7195
7196                                ;      SETUP STATE INDICATOR FOR RETRY ATTEMPT TO NEXT RATE
7197                                ;       00000000B (500) -> 10000000B    (250)
7198                                ;       10000000B (250) -> 01000000B    (300)
7199                                ;       01000000B (300) -> 00000000B    (500)
7200
7201 0000282D 80FC01                       CMP    AH,RATE_500+1       ; SET CY FOR RATE 500
7202 00002830 D0DC                         RCR    AH,1                ; TO NEXT STATE
7203 00002832 80E4C0                       AND    AH,RATE_MSK         ; KEEP ONLY RATE BITS
7204 00002835 80A7[D31D0000]1F             AND    byte [DSK_STATE+eDI], ~(RATE_MSK+DBL_STEP)
7205                                                                   ; RATE, DBL STEP OFF
7206 0000283C 08A7[D31D0000]               OR     [DSK_STATE+eDI],AH  ; TURN ON NEW RATE
7207 00002842 C605[C51D0000]00             MOV    byte [DSKETTE_STATUS],0   ; RESET STATUS FOR RETRY
7208 00002849 F9                           STC                        ; SET CARRY FOR RETRY
7209 0000284A C3                           RETn                       ; RETRY RETURN
7210
7211                                NO_RETRY:
7212 0000284B F8                           CLC                        ; CLEAR CARRY NO RETRY
7213 0000284C C3                           RETn                       ; NO RETRY RETURN
7214
7215                                ;--------------------------------------------------------------------------
7216                                ; NUM_TRANS
7217                                ;     THIS ROUTINE CALCULATES THE NUMBER OF SECTORS THAT WERE
7218                                ;     ACTUALLY TRANSFERRED TO/FROM THE DISKETTE.
7219                                ;
7220                                ; ON ENTRY:  [BP+1] = TRACK
7221                                ;            SI-HI  = HEAD
7222                                ;            [BP]   = START SECTOR
```

```
7223                                    ;
7224                                    ; ON EXIT:  AL = NUMBER ACTUALLY TRANSFERRED
7225                                    ;--------------------------------------------------------------------------
7226                                    NUM_TRANS:
7227 0000284D 30C0                           XOR    AL,AL               ; CLEAR FOR ERROR
7228 0000284F 803D[C51D0000]00               CMP    byte [DSKETTE_STATUS],0   ; CHECK FOR ERROR
7229 00002856 752C                           JNZ    NT_OUT              ; IF ERROR 0 TRANSFERRED
7230 00002858 B204                           MOV    DL,4                ; SECTORS/TRACK OFFSET TO DL
7231 0000285A E8FA000000                     CALL   GET_PARM            ; AH = SECTORS/TRACK
7232 0000285F 8A1D[CB1D0000]                 MOV    BL, [NEC_STATUS+5]  ; GET ENDING SECTOR
7233 00002865 6689F1                         MOV    CX,SI               ; CH = HEAD # STARTED
7234 00002868 3A2D[CA1D0000]                 CMP    CH, [NEC_STATUS+4]  ; GET HEAD ENDED UP ON
7235 0000286E 750D                           JNZ    DIF_HD              ; IF ON SAME HEAD, THEN NO ADJUST
7236 00002870 8A2D[C91D0000]                 MOV    CH, [NEC_STATUS+3]  ; GET TRACK ENDED UP ON
7237 00002876 3A6D01                         CMP    CH,[eBP+1]          ; IS IT ASKED FOR TRACK
7238 00002879 7404                           JZ     short SAME_TRK            ; IF SAME TRACK NO INCREASE
7239 0000287B 00E3                           ADD    BL,AH               ; ADD SECTORS/TRACK
7240                                    DIF_HD:
7241 0000287D 00E3                           ADD    BL,AH               ; ADD SECTORS/TRACK
7242                                    SAME_TRK:
7243 0000287F 2A5D00                         SUB    BL,[eBP]            ; SUBTRACT START FROM END
7244 00002882 88D8                           MOV    AL,BL               ; TO AL
7245                                    NT_OUT:
7246 00002884 C3                             RETn
7247
7248                                    ;--------------------------------------------------------------------------
7249                                    ; SETUP_END
7250                                    ;     RESTORES @MOTOR_COUNT TO PARAMETER PROVIDED IN TABLE
7251                                    ;     AND LOADS @DSKETTE_STATUS TO AH, AND SETS CY.
7252                                    ;
7253                                    ; ON EXIT:
7254                                    ;     AH, @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
7255                                    ;--------------------------------------------------------------------------
7256                                    SETUP_END:
7257 00002885 B202                           MOV    DL,2                ; GET THE MOTOR WAIT PARAMETER
7258 00002887 6650                           PUSH   AX                  ; SAVE NUMBER TRANSFERRED
7259 00002889 E8CB000000                     CALL   GET_PARM
7260 0000288E 8825[C41D0000]                 MOV    [MOTOR_COUNT],AH    ; STORE UPON RETURN
7261 00002894 6658                           POP    AX                  ; RESTORE NUMBER TRANSFERRED
7262 00002896 8A25[C51D0000]                 MOV    AH, [DSKETTE_STATUS]      ; GET STATUS OF OPERATION
7263 0000289C 08E4                           OR     AH,AH               ; CHECK FOR ERROR
7264 0000289E 7402                           JZ     short NUN_ERR             ; NO ERROR
7265 000028A0 30C0                           XOR    AL,AL               ; CLEAR NUMBER RETURNED
7266                                    NUN_ERR:
7267 000028A2 80FC01                         CMP    AH,1                ; SET THE CARRY FLAG TO INDICATE
7268 000028A5 F5                             CMC                        ; SUCCESS OR FAILURE
7269 000028A6 C3                             RETn
7270
7271                                    ;--------------------------------------------------------------------------
```

```
                                ; SETUP_DBL
                                ;     CHECK DOUBLE STEP.
                                ;
                                ; ON ENTRY : DI = DRIVE
                                ;
                                ; ON EXIT :  CY = 1 MEANS ERROR
                                ;--------------------------------------------------------------------------------
                                SETUP_DBL:
000028A7 8AA7[D31D0000]               MOV    AH, [DSK_STATE+eDI] ; ACCESS STATE
000028AD F6C410                       TEST   AH,MED_DET          ; ESTABLISHED STATE ?
000028B0 757E                         JNZ    short NO_DBL              ; IF ESTABLISHED THEN DOUBLE DONE

                                ;-----      CHECK FOR TRACK 0 TO SPEED UP ACKNOWLEDGE OF UNFORMATTED DISKETTE

000028B2 C605[C21D0000]00             MOV    byte [SEEK_STATUS],0      ; SET RECALIBRATE REQUIRED ON ALL DRIVES
000028B9 E8E0000000                   CALL   MOTOR_ON           ; ENSURE MOTOR STAY ON
000028BE B500                         MOV    CH,0               ; LOAD TRACK 0
000028C0 E8C9010000                   CALL   SEEK               ; SEEK TO TRACK 0
000028C5 E868000000                   CALL   READ_ID                 ; READ ID FUNCTION
000028CA 7249                         JC     short SD_ERR       ; IF ERROR NO TRACK 0

                                ;-----      INITIALIZE START AND MAX TRACKS (TIMES 2 FOR BOTH HEADS)

000028CC 66B95004                     MOV    CX,0450H           ; START, MAX TRACKS
000028D0 F687[D31D0000]01             TEST   byte [DSK_STATE+eDI],TRK_CAPA ; TEST FOR 80 TRACK CAPABILITY
000028D7 7402                         JZ     short CNT_OK       ; IF NOT COUNT IS SETUP
000028D9 B1A0                         MOV    CL,0A0H                 ; MAXIMUM TRACK 1.2 MB

                                ;     ATTEMPT READ ID OF ALL TRACKS, ALL HEADS UNTIL SUCCESS; UPON SUCCESS,
                                ;     MUST SEE IF ASKED FOR TRACK IN SINGLE STEP MODE = TRACK ID READ; IF NOT
                                ;     THEN SET DOUBLE STEP ON.

                                CNT_OK:
000028DB C605[C41D0000]FF               MOV     byte [MOTOR_COUNT], 0FFH ; ENSURE MOTOR STAYS ON FOR OPERATION
000028E2 6651                         PUSH   CX                 ; SAVE TRACK, COUNT
000028E4 C605[C51D0000]00             MOV    byte [DSKETTE_STATUS],0   ; CLEAR STATUS, EXPECT ERRORS
000028EB 6631C0                       XOR    AX,AX              ; CLEAR AX
000028EE D0ED                         SHR    CH,1               ; HALVE TRACK, CY = HEAD
000028F0 C0D003                       RCL    AL,3               ; AX = HEAD IN CORRECT BIT
000028F3 6650                         PUSH   AX                 ; SAVE HEAD
000028F5 E894010000                   CALL   SEEK               ; SEEK TO TRACK
000028FA 6658                         POP    AX                 ; RESTORE HEAD
000028FC 6609C7                       OR     DI,AX              ; DI = HEAD OR'ED DRIVE
000028FF E82E000000                   CALL   READ_ID                 ; READ ID HEAD 0
00002904 9C                           PUSHF                     ; SAVE RETURN FROM READ_ID
00002905 6681E7FB00                   AND    DI,11111011B       ; TURN OFF HEAD 1 BIT
0000290A 9D                           POPF                      ; RESTORE ERROR RETURN
0000290B 6659                         POP    CX                 ; RESTORE COUNT
0000290D 7308                         JNC    short DO_CHK       ; IF OK, ASKED = RETURNED TRACK ?
```

```
7321 0000290F FEC5                      INC    CH                  ; INC FOR NEXT TRACK
7322 00002911 38CD                      CMP    CH,CL               ; REACHED MAXIMUM YET
7323 00002913 75C6                      JNZ    short CNT_OK        ; CONTINUE TILL ALL TRIED
7324
7325                                ;-----      FALL THRU, READ ID FAILED FOR ALL TRACKS
7326
7327                                SD_ERR:
7328 00002915 F9                        STC                        ; SET CARRY FOR ERROR
7329 00002916 C3                        RETn                       ; SETUP_DBL ERROR EXIT
7330
7331                                DO_CHK:
7332 00002917 8A0D[C91D0000]            MOV    CL, [NEC_STATUS+3]  ; LOAD RETURNED TRACK
7333 0000291D 888F[D71D0000]            MOV    [DSK_TRK+eDI], CL   ; STORE TRACK NUMBER
7334 00002923 D0ED                      SHR    CH,1                ; HALVE TRACK
7335 00002925 38CD                      CMP    CH,CL               ; IS IT THE SAME AS ASKED FOR TRACK
7336 00002927 7407                      JZ     short NO_DBL        ; IF SAME THEN NO DOUBLE STEP
7337 00002929 808F[D31D0000]20          OR     byte [DSK_STATE+eDI],DBL_STEP ; TURN ON DOUBLE STEP REQUIRED
7338                                NO_DBL:
7339 00002930 F8                        CLC                        ; CLEAR ERROR FLAG
7340 00002931 C3                        RETn
7341
7342                                ;--------------------------------------------------------------------------
7343                                ; READ_ID
7344                                ;     READ ID FUNCTION.
7345                                ;
7346                                ; ON ENTRY:  DI : BIT 2 = HEAD; BITS 1,0 = DRIVE
7347                                ;
7348                                ; ON EXIT:   DI : BIT 2 IS RESET, BITS 1,0 = DRIVE
7349                                ;            @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION
7350                                ;--------------------------------------------------------------------------
7351                                READ_ID:
7352 00002932 B8[4F290000]              MOV    eAX, ER_3           ; MOVE NEC OUTPUT ERROR ADDRESS
7353 00002937 50                        PUSH   eAX
7354 00002938 B44A                      MOV    AH,4AH              ; READ ID COMMAND
7355 0000293A E820010000                CALL   NEC_OUTPUT          ; TO CONTROLLER
7356 0000293F 6689F8                    MOV    AX,DI               ; DRIVE # TO AH, HEAD 0
7357 00002942 88C4                      MOV    AH,AL
7358 00002944 E816010000                CALL   NEC_OUTPUT          ; TO CONTROLLER
7359 00002949 E80BFEFFFF                CALL   NEC_TERM            ; WAIT FOR OPERATION, GET STATUS
7360 0000294E 58                        POP    eAX                 ; THROW AWAY ERROR ADDRESS
7361                                ER_3:
7362 0000294F C3                        RETn
7363
7364                                ;--------------------------------------------------------------------------
7365                                ; CMOS_TYPE
7366                                ;     RETURNS DISKETTE TYPE FROM CMOS
7367                                ;
7368                                ; ON ENTRY:  DI = DRIVE #
7369                                ;
```

```
                                ; ON EXIT:  AL = TYPE; CY REFLECTS STATUS
                                ;--------------------------------------------------------------------------------

                                CMOS_TYPE: ; 11/12/2014
00002950 8A87[16490000]         mov   al, [eDI+fd0_type]
00002956 20C0                   and   al, al ; 18/12/2014
00002958 C3                     retn

                                ;CMOS_TYPE:
                                ;     MOV   AL, CMOS_DIAG       ; CMOS DIAGNOSTIC STATUS BYTE ADDRESS
                                ;     CALL  CMOS_READ           ; GET CMOS STATUS
                                ;     TEST  AL,BAD_BAT+BAD_CKSUM      ; BATTERY GOOD AND CHECKSUM VALID
                                ;     STC                       ; SET CY = 1 INDICATING ERROR FOR RETURN
                                ;     JNZ   short BAD_CM        ; ERROR IF EITHER BIT ON
                                ;     MOV   AL,CMOS_DISKETTE    ; ADDRESS OF DISKETTE BYTE IN CMOS
                                ;     CALL  CMOS_READ           ; GET DISKETTE BYTE
                                ;     OR    DI,DI               ; SEE WHICH DRIVE IN QUESTION
                                ;     JNZ   short TB            ; IF DRIVE 1, DATA IN LOW NIBBLE
                                ;     ROR   AL,4                ; EXCHANGE NIBBLES IF SECOND DRIVE
                                ;TB:
                                ;     AND   AL,0FH              ; KEEP ONLY DRIVE DATA, RESET CY, 0
                                ;BAD_CM:
                                ;     RETn                      ; CY, STATUS OF READ

                                ;--------------------------------------------------------------------------------
                                ; GET_PARM
                                ;     THIS ROUTINE FETCHES THE INDEXED POINTER FROM THE DISK_BASE
                                ;     BLOCK POINTED TO BY THE DATA VARIABLE @DISK_POINTER. A BYTE FROM
                                ;     THAT TABLE IS THEN MOVED INTO AH, THE INDEX OF THAT BYTE BEING
                                ;     THE PARAMETER IN DL.
                                ;
                                ; ON ENTRY:  DL = INDEX OF BYTE TO BE FETCHED
                                ;
                                ; ON EXIT:   AH = THAT BYTE FROM BLOCK
                                ;           AL,DH DESTROYED
                                ;--------------------------------------------------------------------------------
                                GET_PARM:
                                      ;PUSH  DS
00002959 56                           PUSH   eSI
                                      ;SUB   AX,AX              ; DS = 0, BIOS DATA AREA
                                      ;MOV   DS,AX
                                      ;;mov  ax, cs
                                      ;;mov  ds, ax
                                      ; 08/02/2015 (protected mode modifications, bx -> ebx)
0000295A 87D3                         XCHG   eDX,eBX                  ; BL = INDEX
                                      ;SUB   BH,BH              ; BX = INDEX
0000295C 81E3FF000000                 and    ebx, 0FFh
                                      ;LDS   SI, [DISK_POINTER]  ; POINT TO BLOCK
                                      ;
```

```
7419                                    ; 17/12/2014
7420 00002962 66A1[D91D0000]            mov    ax, [cfd] ; current (AL) and previous fd (AH)
7421 00002968 38E0                      cmp    al, ah
7422 0000296A 7425                      je     short gpndc
7423 0000296C A2[DA1D0000]              mov    [pfd], al ; current drive -> previous drive
7424 00002971 53                        push   ebx ; 08/02/2015
7425 00002972 88C3                      mov    bl, al
7426                                    ; 11/12/2014
7427 00002974 8A83[16490000]            mov    al, [eBX+fd0_type]  ; Drive type (0,1,2,3,4)
7428                                    ; 18/12/2014
7429 0000297A 20C0                      and    al, al
7430 0000297C 7507                      jnz    short gpdtc
7431 0000297E BB[3A1E0000]              mov    ebx, MD_TBL6        ; 1.44 MB param. tbl. (default)
7432 00002983 EB05                        jmp     short gpdpu
7433                                  gpdtc:
7434 00002985 E816F9FFFF                call   DR_TYPE_CHECK
7435                                    ; cf = 1 -> eBX points to 1.44MB fd parameter table (default)
7436                                  gpdpu:
7437 0000298A 891D[BE1D0000]            mov    [DISK_POINTER], ebx
7438 00002990 5B                        pop    ebx
7439                                  gpndc:
7440 00002991 8B35[BE1D0000]            mov    esi, [DISK_POINTER] ; 08/02/2015, si -> esi
7441 00002997 8A241E                    MOV    AH, [eSI+eBX]       ; GET THE WORD
7442 0000299A 87D3                      XCHG   eDX,eBX                   ; RESTORE BX
7443 0000299C 5E                        POP    eSI
7444                                    ;POP   DS
7445 0000299D C3                        RETn
7446
7447                                  ;--------------------------------------------------------------------------
7448                                  ; MOTOR_ON
7449                                  ;     TURN MOTOR ON AND WAIT FOR MOTOR START UP TIME. THE @MOTOR_COUNT
7450                                  ;     IS REPLACED WITH A SUFFICIENTLY HIGH NUMBER (0FFH) TO ENSURE
7451                                  ;     THAT THE MOTOR DOES NOT GO OFF DURING THE OPERATION. IF THE
7452                                  ;     MOTOR NEEDED TO BE TURNED ON, THE MULTI-TASKING HOOK FUNCTION
7453                                  ;     (AX=90FDH, INT 15) IS CALLED TELLING THE OPERATING SYSTEM
7454                                  ;     THAT THE BIOS IS ABOUT TO WAIT FOR MOTOR START UP. IF THIS
7455                                  ;     FUNCTION RETURNS WITH CY = 1, IT MEANS THAT THE MINIMUM WAIT
7456                                  ;     HAS BEEN COMPLETED. AT THIS POINT A CHECK IS MADE TO ENSURE
7457                                  ;     THAT THE MOTOR WASN'T TURNED OFF BY THE TIMER. IF THE HOOK DID
7458                                  ;     NOT WAIT, THE WAIT FUNCTION (AH=086H) IS CALLED TO WAIT THE
7459                                  ;     PRESCRIBED AMOUNT OF TIME. IF THE CARRY FLAG IS SET ON RETURN,
7460                                  ;     IT MEANS THAT THE FUNCTION IS IN USE AND DID NOT PERFORM THE
7461                                  ;     WAIT. A TIMER 1 WAIT LOOP WILL THEN DO THE WAIT.
7462                                  ;
7463                                  ; ON ENTRY:  DI = DRIVE #
7464                                  ; ON EXIT:   AX,CX,DX DESTROYED
7465                                  ;--------------------------------------------------------------------------
7466                                  MOTOR_ON:
7467 0000299E 53                        PUSH   eBX                ; SAVE REG.
```

```
7468 0000299F E82A000000                CALL   TURN_ON                   ; TURN ON MOTOR
7469 000029A4 7226                      JC     short MOT_IS_ON           ; IF CY=1 NO WAIT
7470 000029A6 E89AF9FFFF                CALL   XLAT_OLD            ; TRANSLATE STATE TO COMPATIBLE MODE
7471 000029AB E864F9FFFF                CALL   XLAT_NEW            ; TRANSLATE STATE TO PRESENT ARCH,
7472                                    ;CALL  TURN_ON             ; CHECK AGAIN IF MOTOR ON
7473                                    ;JC    MOT_IS_ON           ; IF NO WAIT MEANS IT IS ON
7474                                M_WAIT:
7475 000029B0 B20A                      MOV    DL,10               ; GET THE MOTOR WAIT PARAMETER
7476 000029B2 E8A2FFFFFF                CALL   GET_PARM
7477                                    ;MOV   AL,AH               ; AL = MOTOR WAIT PARAMETER
7478                                    ;XOR   AH,AH               ; AX = MOTOR WAIT PARAMETER
7479                                    ;CMP   AL,8                ; SEE IF AT LEAST A SECOND IS SPECIFIED
7480 000029B7 80FC08                    cmp    ah, 8
7481                                    ;JAE   short GP2           ; IF YES, CONTINUE
7482 000029BA 7702                      ja     short J13
7483                                    ;MOV   AL,8                ; ONE SECOND WAIT FOR MOTOR START UP
7484 000029BC B408                      mov    ah, 8
7485
7486                                ;-----      AS CONTAINS NUMBER OF 1/8 SECONDS (125000 MICROSECONDS) TO WAIT
7487                                GP2:
7488                                ;-----      FOLLOWING LOOPS REQUIRED WHEN RTC WAIT FUNCTION IS ALREADY IN USE
7489                                J13:                               ; WAIT FOR 1/8 SECOND PER (AL)
7490 000029BE B95E200000                MOV    eCX,8286            ; COUNT FOR 1/8 SECOND AT 15.085737 US
7491 000029C3 E88DF0FFFF                CALL   WAITF               ; GO TO FIXED WAIT ROUTINE
7492                                    ;DEC   AL                  ; DECREMENT TIME VALUE
7493 000029C8 FECC                      dec    ah
7494 000029CA 75F2                      JNZ    short J13           ; ARE WE DONE YET
7495                                MOT_IS_ON:
7496 000029CC 5B                        POP    eBX                 ; RESTORE REG.
7497 000029CD C3                        RETn
7498
7499                                ;--------------------------------------------------------------------------------
7500                                ; TURN_ON
7501                                ;     TURN MOTOR ON AND RETURN WAIT STATE.
7502                                ;
7503                                ; ON ENTRY:  DI = DRIVE #
7504                                ;
7505                                ; ON EXIT:   CY = 0 MEANS WAIT REQUIRED
7506                                ;            CY = 1 MEANS NO WAIT REQUIRED
7507                                ;            AX,BX,CX,DX DESTROYED
7508                                ;--------------------------------------------------------------------------------
7509                                TURN_ON:
7510 000029CE 89FB                      MOV    eBX,eDI                   ; BX = DRIVE #
7511 000029D0 88D9                      MOV    CL,BL               ; CL = DRIVE #
7512 000029D2 C0C304                    ROL    BL,4                ; BL = DRIVE SELECT
7513 000029D5 FA                        CLI                        ; NO INTERRUPTS WHILE DETERMINING STATUS
7514 000029D6 C605[C41D0000]FF          MOV    byte [MOTOR_COUNT],0FFH   ; ENSURE MOTOR STAYS ON FOR OPERATION
7515 000029DD A0[C31D0000]              MOV    AL, [MOTOR_STATUS]  ; GET DIGITAL OUTPUT REGISTER REFLECTION
7516 000029E2 2430                      AND    AL,00110000B        ; KEEP ONLY DRIVE SELECT BITS
```

```
7517 000029E4 B401                      MOV    AH,1               ; MASK FOR DETERMINING MOTOR BIT
7518 000029E6 D2E4                      SHL    AH,CL              ; AH = MOTOR ON, A=00000001, B=00000010
7519
7520                                 ;  AL = DRIVE SELECT FROM @MOTOR_STATUS
7521                                 ;  BL = DRIVE SELECT DESIRED
7522                                 ;  AH = MOTOR ON MASK DESIRED
7523
7524 000029E8 38D8                      CMP    AL,BL              ; REQUESTED DRIVE ALREADY SELECTED ?
7525 000029EA 7508                      JNZ    short TURN_IT_ON   ; IF NOT SELECTED JUMP
7526 000029EC 8425[C31D0000]            TEST   AH, [MOTOR_STATUS] ; TEST MOTOR ON BIT
7527 000029F2 7535                      JNZ    short NO_MOT_WAIT  ; JUMP IF MOTOR ON AND SELECTED
7528
7529                                 TURN_IT_ON:
7530 000029F4 08DC                      OR     AH,BL              ; AH = DRIVE SELECT AND MOTOR ON
7531 000029F6 8A3D[C31D0000]            MOV    BH, [MOTOR_STATUS] ; SAVE COPY OF @MOTOR_STATUS BEFORE
7532 000029FC 80E70F                    AND    BH,00001111B       ; KEEP ONLY MOTOR BITS
7533 000029FF 8025[C31D0000]CF          AND    byte [MOTOR_STATUS],11001111B ; CLEAR OUT DRIVE SELECT
7534 00002A06 0825[C31D0000]            OR     byte [MOTOR_STATUS],AH    ; OR IN DRIVE SELECTED AND MOTOR ON
7535 00002A0C A0[C31D0000]              MOV    AL, [MOTOR_STATUS] ; GET DIGITAL OUTPUT REGISTER REFLECTION
7536 00002A11 88C3                      MOV    BL,AL              ; BL=@MOTOR_STATUS AFTER, BH=BEFORE
7537 00002A13 80E30F                    AND    BL,00001111B       ; KEEP ONLY MOTOR BITS
7538 00002A16 FB                        STI                       ; ENABLE INTERRUPTS AGAIN
7539 00002A17 243F                      AND    AL,00111111B       ; STRIP AWAY UNWANTED BITS
7540 00002A19 C0C004                    ROL    AL,4               ; PUT BITS IN DESIRED POSITIONS
7541 00002A1C 0C0C                      OR     AL,00001100B       ; NO RESET, ENABLE DMA/INTERRUPT
7542 00002A1E 66BAF203                  MOV    DX,03F2H           ; SELECT DRIVE AND TURN ON MOTOR
7543 00002A22 EE                        OUT    DX,AL
7544 00002A23 38FB                      CMP    BL,BH              ; NEW MOTOR TURNED ON ?
7545 00002A25 7402                      JZ     short NO_MOT_WAIT  ; NO WAIT REQUIRED IF JUST SELECT
7546 00002A27 F8                        CLC                       ; (re)SET CARRY MEANING WAIT
7547 00002A28 C3                        RETn
7548
7549                                 NO_MOT_WAIT:
7550 00002A29 F9                        STC                       ; SET NO WAIT REQUIRED
7551 00002A2A FB                        STI                       ; INTERRUPTS BACK ON
7552 00002A2B C3                        RETn
7553
7554                                 ;--------------------------------------------------------------------------------
7555                                 ; HD_WAIT
7556                                 ;     WAIT FOR HEAD SETTLE TIME.
7557                                 ;
7558                                 ; ON ENTRY:  DI = DRIVE #
7559                                 ;
7560                                 ; ON EXIT:   AX,BX,CX,DX DESTROYED
7561                                 ;--------------------------------------------------------------------------------
7562                                 HD_WAIT:
7563 00002A2C B209                      MOV    DL,9               ; GET HEAD SETTLE PARAMETER
7564 00002A2E E826FFFFFF                CALL   GET_PARM
7565 00002A33 08E4                      or     ah, ah ; 17/12/2014
```

```
7566 00002A35 7519                      jnz    short DO_WAT
7567 00002A37 F605[C31D0000]80            TEST    byte [MOTOR_STATUS],10000000B ; SEE IF A WRITE OPERATION
7568                                    ;JZ    short ISNT_WRITE    ; IF NOT, DO NOT ENFORCE ANY VALUES
7569                                    ;OR    AH,AH               ; CHECK FOR ANY WAIT?
7570                                    ;JNZ   short DO_WAT        ; IF THERE DO NOT ENFORCE
7571 00002A3E 741E                      jz     short HW_DONE
7572 00002A40 B40F                      MOV    AH,HD12_SETTLE            ; LOAD 1.2M HEAD SETTLE MINIMUM
7573 00002A42 8A87[D31D0000]            MOV    AL,[DSK_STATE+eDI]  ; LOAD STATE
7574 00002A48 24C0                      AND    AL,RATE_MSK         ; KEEP ONLY RATE
7575 00002A4A 3C80                      CMP    AL,RATE_250         ; 1.2 M DRIVE ?
7576 00002A4C 7502                      JNZ    short DO_WAT        ; DEFAULT HEAD SETTLE LOADED
7577                                ;GP3:
7578 00002A4E B414                      MOV    AH,HD320_SETTLE           ; USE 320/360 HEAD SETTLE
7579                                ;     JMP    SHORT DO_WAT
7580
7581                                ;ISNT_WRITE:
7582                                ;     OR     AH,AH               ; CHECK FOR NO WAIT
7583                                ;     JZ     short HW_DONE       ; IF NOT WRITE AND 0 ITS OK
7584
7585                                ;-----       AH CONTAINS NUMBER OF MILLISECONDS TO WAIT
7586                                DO_WAT:
7587                                ;     MOV    AL,AH               ; AL = # MILLISECONDS
7588                                ;     ;XOR   AH,AH               ; AX = # MILLISECONDS
7589                                J29:                              ;     1 MILLISECOND LOOP
7590                                      ;mov   cx, WAIT_FDU_HEAD_SETTLE ; 33 ; 1 ms in 30 micro units.
7591 00002A50 B942000000                 MOV    eCX,66              ; COUNT AT 15.085737 US PER COUNT
7592 00002A55 E8FBEFFFFF                 CALL   WAITF               ; DELAY FOR 1 MILLISECOND
7593                                      ;DEC   AL                  ; DECREMENT THE COUNT
7594 00002A5A FECC                       dec    ah
7595 00002A5C 75F2                       JNZ    short J29           ; DO AL MILLISECOND # OF TIMES
7596                                HW_DONE:
7597 00002A5E C3                         RETn
7598
7599                                ;--------------------------------------------------------------------------------
7600                                ; NEC_OUTPUT
7601                                ;     THIS ROUTINE SENDS A BYTE TO THE NEC CONTROLLER AFTER TESTING
7602                                ;     FOR CORRECT DIRECTION AND CONTROLLER READY THIS ROUTINE WILL
7603                                ;     TIME OUT IF THE BYTE IS NOT ACCEPTED WITHIN A REASONABLE AMOUNT
7604                                ;     OF TIME, SETTING THE DISKETTE STATUS ON COMPLETION.
7605                                ;
7606                                ; ON ENTRY:  AH = BYTE TO BE OUTPUT
7607                                ;
7608                                ; ON EXIT:   CY = 0  SUCCESS
7609                                ;            CY = 1  FAILURE -- DISKETTE STATUS UPDATED
7610                                ;                    IF A FAILURE HAS OCCURRED, THE RETURN IS MADE ONE LEVEL
7611                                ;                    HIGHER THAN THE CALLER OF NEC OUTPUT. THIS REMOVES THE
7612                                ;                    REQUIREMENT OF TESTING AFTER EVERY CALL OF NEC_OUTPUT.
7613                                ;            AX,CX,DX DESTROYED
7614                                ;--------------------------------------------------------------------------------
```

```
7615
7616                                    ; 09/12/2014 [Erdogan Tan]
7617                                    ;     (from 'PS2 Hardware Interface Tech. Ref. May 88', Page 09-05.)
7618                                    ; Diskette Drive Controller Status Register (3F4h)
7619                                    ;     This read only register facilitates the transfer of data between
7620                                    ;     the system microprocessor and the controller.
7621                                    ; Bit 7 - When set to 1, the Data register is ready to transfer data
7622                                    ;       with the system micrprocessor.
7623                                    ; Bit 6 - The direction of data transfer. If this bit is set to 0,
7624                                    ;       the transfer is to the controller.
7625                                    ; Bit 5 - When this bit is set to 1, the controller is in the non-DMA mode.
7626                                    ; Bit 4 - When this bit is set to 1, a Read or Write command is being executed.
7627                                    ; Bit 3 - Reserved.
7628                                    ; Bit 2 - Reserved.
7629                                    ; Bit 1 - When this bit is set to 1, dskette drive 1 is in the seek mode.
7630                                    ; Bit 0 - When this bit is set to 1, dskette drive 1 is in the seek mode.
7631
7632                                    ; Data Register (3F5h)
7633                                    ; This read/write register passes data, commands and parameters, and provides
7634                                    ; diskette status information.
7635
7636                                    NEC_OUTPUT:
7637                                          ;PUSH  BX                 ; SAVE REG.
7638 00002A5F 66BAF403                       MOV   DX,03F4H            ; STATUS PORT
7639                                          ;MOV   BL,2               ; HIGH ORDER COUNTER
7640                                          ;XOR   CX,CX              ; COUNT FOR TIME OUT
7641                                          ; 16/12/2014
7642                                          ; waiting for (max.) 0.5 seconds
7643 00002A63 C605[2A480000]00                 mov    byte [wait_count], 0
7644                                          ;
7645                                          ; 17/12/2014
7646                                          ; Modified from AWARD BIOS 1999 - ADISK.ASM - SEND_COMMAND
7647                                          ;
7648                                          ;WAIT_FOR_PORT:    Waits for a bit at a port pointed to by DX to
7649                                          ;           go on.
7650                                          ;INPUT:
7651                                          ;     AH=Mask for isolation bits.
7652                                          ;     AL=pattern to look for.
7653                                          ;     DX=Port to test for
7654                                          ;     BH:CX=Number of memory refresh periods to delay.
7655                                          ;          (normally 30 microseconds per period.)
7656                                          ;
7657                                          ;WFP_SHORT:
7658                                          ;     Wait for port if refresh cycle is short (15-80 Us range).
7659                                          ;
7660
7661                                    ;     mov   bl, WAIT_FDU_SEND_HI+1    ; 0+1
7662                                    ;     mov   cx, WAIT_FDU_SEND_LO      ; 16667
7663                                    ;
```

```
                                ;WFPS_OUTER_LP:
                                ;     ;
                                ;WFPS_CHECK_PORT:
                                ;;J23:
                                ;     IN    AL,DX               ; GET STATUS
                                ;     AND   AL,11000000B        ; KEEP STATUS AND DIRECTION
                                ;     CMP   AL,10000000B        ; STATUS 1 AND DIRECTION 0 ?
                                ;     JZ    short J27           ; STATUS AND DIRECTION OK
                                ;WFPS_HI:
                                ;     IN    AL, PORT_B   ;061h  ; SYS1 ; wait for hi to lo
                                ;     TEST  AL,010H                   ; transition on memory
                                ;     JNZ   SHORT WFPS_HI       ; refresh.
                                ;WFPS_LO:
                                ;     IN    AL, PORT_B          ; SYS1
                                ;     TEST  AL,010H
                                ;     JZ    SHORT WFPS_LO
                                ;     LOOP  SHORT WFPS_CHECK_PORT
                                ;     ;
                                ;     dec   bl
                                ;     jnz   short WFPS_OUTER_LP
                                ;     jmp   short WFPS_TIMEOUT  ; fail

                                J23:
00002A6A EC                           IN    AL,DX               ; GET STATUS
00002A6B 24C0                         AND   AL,11000000B        ; KEEP STATUS AND DIRECTION
00002A6D 3C80                         CMP   AL,10000000B        ; STATUS 1 AND DIRECTION 0 ?
00002A6F 7413                         JZ    short J27           ; STATUS AND DIRECTION OK
                                      ;LOOP  J23                ; CONTINUE TILL CX EXHAUSTED
                                      ;DEC   BL                 ; DECREMENT COUNTER
                                      ;JNZ   short J23          ; REPEAT TILL DELAY FINISHED, CX = 0

                                      ;16/12/2014
00002A71 803D[2A480000]0A               cmp     byte [wait_count], 10   ; (10/18.2 seconds)
00002A78 72F0                         jb    short J23

                                ;WFPS_TIMEOUT:

                                ;-----       FALL THRU TO ERROR RETURN

00002A7A 800D[C51D0000]80             OR    byte [DSKETTE_STATUS],TIME_OUT
                                      ;POP   BX                 ; RESTORE REG.
00002A81 58                           POP   eAX ; 08/02/2015    ; DISCARD THE RETURN ADDRESS
00002A82 F9                           STC                       ; INDICATE ERROR TO CALLER
00002A83 C3                           RETn

                                ;-----       DIRECTION AND STATUS OK; OUTPUT BYTE

                                J27:
00002A84 88E0                         MOV   AL,AH               ; GET BYTE TO OUTPUT
```

```
7713 00002A86 6642                          INC    DX                ; DATA PORT = STATUS PORT + 1
7714 00002A88 EE                            OUT    DX,AL             ; OUTPUT THE BYTE
7715                                        IODELAY
7716 00002A89 EB00                  <1>  jmp short $+2
7717 00002A8B EB00                  <1>  jmp short $+2
7718                                        ;
7719                                        ;PUSHF                   ; SAVE FLAGS
7720                                        ;MOV   CX, 3             ; 30 TO 45 MICROSECONDS WAIT FOR
7721                                        ;CALL  WAITF             ; NEC FLAGS UPDATE CYCLE
7722                                        ;POPF                    ; RESTORE FLAGS FOR EXIT
7723                                        ;POP   BX                ; RESTORE REG
7724 00002A8D C3                            RETn                     ; CY = 0 FROM TEST INSTRUCTION
7725
7726                                    ;--------------------------------------------------------------------------
7727                                    ; SEEK
7728                                    ;     THIS ROUTINE WILL MOVE THE HEAD ON THE NAMED DRIVE TO THE NAMED
7729                                    ;     TRACK. IF THE DRIVE HAS NOT BEEN ACCESSED SINCE THE DRIVE
7730                                    ;     RESET COMMAND WAS ISSUED, THE DRIVE WILL BE RECALIBRATED.
7731                                    ;
7732                                    ; ON ENTRY:  DI = DRIVE #
7733                                    ;            CH = TRACK #
7734                                    ;
7735                                    ; ON EXIT:   @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.
7736                                    ;            AX,BX,CX DX DESTROYED
7737                                    ;--------------------------------------------------------------------------
7738                                    SEEK:
7739 00002A8E 89FB                          MOV    eBX,eDI                  ; BX = DRIVE #
7740 00002A90 B001                          MOV    AL,1              ; ESTABLISH MASK FOR RECALIBRATE TEST
7741 00002A92 86CB                          XCHG   CL,BL             ; SET DRIVE VALULE INTO CL
7742 00002A94 D2C0                          ROL    AL,CL             ; SHIFT MASK BY THE DRIVE VALUE
7743 00002A96 86CB                          XCHG   CL,BL             ; RECOVER TRACK VALUE
7744 00002A98 8405[C21D0000]                TEST   AL,[SEEK_STATUS]  ; TEST FOR RECALIBRATE REQUIRED
7745 00002A9E 7526                          JNZ    short J28A        ; JUMP IF RECALIBRATE NOT REQUIRED
7746
7747 00002AA0 0805[C21D0000]                OR     [SEEK_STATUS],AL  ; TURN ON THE NO RECALIBRATE BIT IN FLAG
7748 00002AA6 E862000000                    CALL   RECAL             ; RECALIBRATE DRIVE
7749 00002AAB 730E                          JNC    short AFT_RECAL          ; RECALIBRATE DONE
7750
7751                                    ;-----      ISSUE RECALIBRATE FOR 80 TRACK DISKETTES
7752
7753 00002AAD C605[C51D0000]00              MOV    byte [DSKETTE_STATUS],0   ; CLEAR OUT INVALID STATUS
7754 00002AB4 E854000000                    CALL   RECAL             ; RECALIBRATE DRIVE
7755 00002AB9 7251                          JC     short RB          ; IF RECALIBRATE FAILS TWICE THEN ERROR
7756
7757                                    AFT_RECAL:
7758 00002ABB C687[D71D0000]00                MOV     byte [DSK_TRK+eDI],0    ; SAVE NEW CYLINDER AS PRESENT POSITION
7759 00002AC2 08ED                          OR     CH,CH             ; CHECK FOR SEEK TO TRACK 0
7760 00002AC4 743F                          JZ     short DO_WAIT     ; HEAD SETTLE, CY = 0 IF JUMP
7761
```

```
7762                                    ;-----      DRIVE IS IN SYNCHRONIZATION WITH CONTROLLER, SEEK TO TRACK
7763
7764 00002AC6 F687[D31D0000]20          J28A: TEST  byte [DSK_STATE+eDI],DBL_STEP ; CHECK FOR DOUBLE STEP REQUIRED
7765 00002ACD 7402                            JZ    short R7           ; SINGLE STEP REQUIRED BYPASS DOUBLE
7766 00002ACF D0E5                            SHL   CH,1               ; DOUBLE NUMBER OF STEP TO TAKE
7767
7768 00002AD1 3AAF[D71D0000]           R7:    CMP   CH, [DSK_TRK+eDI]  ; SEE IF ALREADY AT THE DESIRED TRACK
7769 00002AD7 7433                            JE    short RB           ; IF YES, DO NOT NEED TO SEEK
7770
7771 00002AD9 BA[0C2B0000]                    MOV   eDX, NEC_ERR       ; LOAD RETURN ADDRESS
7772 00002ADE 52                              PUSH  eDX ; (*)          ; ON STACK FOR NEC OUTPUT ERROR
7773 00002ADF 88AF[D71D0000]                  MOV   [DSK_TRK+eDI],CH   ; SAVE NEW CYLINDER AS PRESENT POSITION
7774 00002AE5 B40F                            MOV   AH,0FH             ; SEEK COMMAND TO NEC
7775 00002AE7 E873FFFFFF                      CALL  NEC_OUTPUT
7776 00002AEC 89FB                            MOV   eBX,eDI                 ; BX = DRIVE #
7777 00002AEE 88DC                            MOV   AH,BL              ; OUTPUT DRIVE NUMBER
7778 00002AF0 E86AFFFFFF                      CALL  NEC_OUTPUT
7779 00002AF5 8AA7[D71D0000]                  MOV   AH, [DSK_TRK+eDI]  ; GET CYLINDER NUMBER
7780 00002AFB E85FFFFFFF                      CALL  NEC_OUTPUT
7781 00002B00 E829000000                      CALL  CHK_STAT_2         ; ENDING INTERRUPT AND SENSE STATUS
7782
7783                                    ;-----      WAIT FOR HEAD SETTLE
7784
7785                                    DO_WAIT:
7786 00002B05 9C                              PUSHF                    ; SAVE STATUS
7787 00002B06 E821FFFFFF                      CALL  HD_WAIT                 ; WAIT FOR HEAD SETTLE TIME
7788 00002B0B 9D                              POPF                     ; RESTORE STATUS
7789                                    RB:
7790                                    NEC_ERR:
7791                                          ; 08/02/2015 (code trick here from original IBM PC/AT DISKETTE.ASM)
7792                                          ; (*) nec_err -> retn (push edx -> pop edx) -> nec_err -> retn
7793 00002B0C C3                              RETn                     ; RETURN TO CALLER
7794
7795                                    ;--------------------------------------------------------------------------------
7796                                    ; RECAL
7797                                    ;     RECALIBRATE DRIVE
7798                                    ;
7799                                    ; ON ENTRY:  DI = DRIVE #
7800                                    ;
7801                                    ; ON EXIT:   CY REFLECTS STATUS OF OPERATION.
7802                                    ;--------------------------------------------------------------------------------
7803                                    RECAL:
7804 00002B0D 6651                            PUSH  CX
7805 00002B0F B8[2B2B0000]                    MOV   eAX, RC_BACK       ; LOAD NEC_OUTPUT ERROR
7806 00002B14 50                              PUSH  eAX
7807 00002B15 B407                            MOV   AH,07H             ; RECALIBRATE COMMAND
7808 00002B17 E843FFFFFF                      CALL  NEC_OUTPUT
7809 00002B1C 89FB                            MOV   eBX,eDI                 ; BX = DRIVE #
7810 00002B1E 88DC                            MOV   AH,BL
```

```
7811 00002B20 E83AFFFFFF                CALL   NEC_OUTPUT          ; OUTPUT THE DRIVE NUMBER
7812 00002B25 E804000000                CALL   CHK_STAT_2          ; GET THE INTERRUPT AND SENSE INT STATUS
7813 00002B2A 58                        POP    eAX                 ; THROW AWAY ERROR
7814                                  RC_BACK:
7815 00002B2B 6659                      POP    CX
7816 00002B2D C3                        RETn
7817
7818                                  ;--------------------------------------------------------------------------
7819                                  ; CHK_STAT_2
7820                                  ;     THIS ROUTINE HANDLES THE INTERRUPT RECEIVED AFTER RECALIBRATE,
7821                                  ;     OR SEEK TO THE ADAPTER. THE INTERRUPT IS WAITED FOR, THE
7822                                  ;     INTERRUPT STATUS SENSED, AND THE RESULT RETURNED TO THE CALLER.
7823                                  ;
7824                                  ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.
7825                                  ;--------------------------------------------------------------------------
7826                                  CHK_STAT_2:
7827 00002B2E B8[562B0000]                MOV     eAX, CS_BACK            ; LOAD NEC_OUTPUT ERROR ADDRESS
7828 00002B33 50                        PUSH   eAX
7829 00002B34 E828000000                CALL   WAIT_INT            ; WAIT FOR THE INTERRUPT
7830 00002B39 721A                      JC     short J34           ; IF ERROR, RETURN IT
7831 00002B3B B408                      MOV    AH,08H              ; SENSE INTERRUPT STATUS COMMAND
7832 00002B3D E81DFFFFFF                CALL   NEC_OUTPUT
7833 00002B42 E847000000                CALL   RESULTS                   ; READ IN THE RESULTS
7834 00002B47 720C                      JC     short J34
7835 00002B49 A0[C61D0000]              MOV    AL,[NEC_STATUS]           ; GET THE FIRST STATUS BYTE
7836 00002B4E 2460                      AND    AL,01100000B        ; ISOLATE THE BITS
7837 00002B50 3C60                      CMP    AL,01100000B        ; TEST FOR CORRECT VALUE
7838 00002B52 7403                      JZ     short J35           ; IF ERROR, GO MARK IT
7839 00002B54 F8                        CLC                        ; GOOD RETURN
7840                                  J34:
7841 00002B55 58                        POP    eAX                 ; THROW AWAY ERROR RETURN
7842                                  CS_BACK:
7843 00002B56 C3                        RETn
7844                                  J35:
7845 00002B57 800D[C51D0000]40          OR     byte [DSKETTE_STATUS], BAD_SEEK
7846 00002B5E F9                        STC                        ; ERROR RETURN CODE
7847 00002B5F EBF4                      JMP    SHORT J34
7848
7849                                  ;--------------------------------------------------------------------------
7850                                  ; WAIT_INT
7851                                  ;     THIS ROUTINE WAITS FOR AN INTERRUPT TO OCCUR A TIME OUT ROUTINE
7852                                  ;     TAKES PLACE DURING THE WAIT, SO THAT AN ERROR MAY BE RETURNED
7853                                  ;     IF THE DRIVE IS NOT READY.
7854                                  ;
7855                                  ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.
7856                                  ;--------------------------------------------------------------------------
7857
7858                                  ; 17/12/2014
7859                                  ; 2.5 seconds waiting !
```

```
                                ;(AWARD BIOS - 1999, WAIT_FDU_INT_LOW, WAIT_FDU_INT_HI)
                                ; amount of time to wait for completion interrupt from NEC.


                                WAIT_INT:
00002B61 FB                           STI                     ; TURN ON INTERRUPTS, JUST IN CASE
00002B62 F8                           CLC                     ; CLEAR TIMEOUT INDICATOR
                                       ;MOV  BL,10            ; CLEAR THE COUNTERS
                                       ;XOR  CX,CX            ; FOR 2 SECOND WAIT

                                      ; Modification from AWARD BIOS - 1999 (ATORGS.ASM, WAIT
                                      ;
                                      ;WAIT_FOR_MEM:
                                      ;     Waits for a bit at a specified memory location pointed
                                      ;     to by ES:[DI] to become set.
                                      ;INPUT:
                                      ;     AH=Mask to test with.
                                      ;     ES:[DI] = memory location to watch.
                                      ;     BH:CX=Number of memory refresh periods to delay.
                                      ;          (normally 30 microseconds per period.)

                                      ; waiting for (max.) 2.5 secs in 30 micro units.
                                ;     mov cx, WAIT_FDU_INT_LO         ; 017798
                                ;;    mov bl, WAIT_FDU_INT_HI
                                ;     mov bl, WAIT_FDU_INT_HI + 1

                                ;WFMS_CHECK_MEM:
                                ;     test   byte [SEEK_STATUS],INT_FLAG ; TEST FOR INTERRUPT OCCURRING
                                ;     jnz    shortr J37
                                ;WFMS_HI:
                                ;     IN     AL,PORT_B  ; 061h   ; SYS1, wait for lo to hi
                                ;     TEST   AL,010H                ; transition on memory
                                ;     JNZ    SHORT WFMS_HI       ; refresh.
                                ;WFMS_LO:
                                ;     IN     AL,PORT_B           ;SYS1
                                ;     TEST   AL,010H
                                ;     JZ     SHORT WFMS_LO
                                ;     LOOP   SHORT WFMS_CHECK_MEM
                                ;WFMS_OUTER_LP:
                                ;;    or     bl, bl              ; check outer counter
                                ;;    jz     short J36A          ; WFMS_TIMEOUT
                                ;     dec    bl
                                ;     jz     short J36A
                                ;     jmp    short WFMS_CHECK_MEM

                                      ;17/12/2014
                                      ;16/12/2014
00002B63 C605[2A480000]00               mov     byte [wait_count], 0    ; Reset (INT 08H) counter
                                J36:
```

```
7909 00002B6A F605[C21D0000]80            TEST   byte [SEEK_STATUS],INT_FLAG ; TEST FOR INTERRUPT OCCURRING
7910 00002B71 7511                        JNZ    short J37
7911                                      ;16/12/2014
7912                                      ;LOOP  J36                 ; COUNT DOWN WHILE WAITING
7913                                      ;DEC   BL                  ; SECOND LEVEL COUNTER
7914                                      ;JNZ   short J36
7915 00002B73 803D[2A480000]2E              cmp     byte [wait_count], 46   ; (46/18.2 seconds)
7916 00002B7A 72EE                        jb     short J36
7917
7918                                    ;WFMS_TIMEOUT:
7919                                    J36A:
7920 00002B7C 800D[C51D0000]80            OR     byte [DSKETTE_STATUS], TIME_OUT ; NOTHING HAPPENED
7921 00002B83 F9                          STC                        ; ERROR RETURN
7922                                    J37:
7923 00002B84 9C                          PUSHF                      ; SAVE CURRENT CARRY
7924 00002B85 8025[C21D0000]7F            AND    byte [SEEK_STATUS], ~INT_FLAG ; TURN OFF INTERRUPT FLAG
7925 00002B8C 9D                          POPF                       ; RECOVER CARRY
7926 00002B8D C3                          RETn                       ; GOOD RETURN CODE
7927
7928                                    ;--------------------------------------------------------------------------
7929                                    ; RESULTS
7930                                    ;     THIS ROUTINE WILL READ ANYTHING THAT THE NEC CONTROLLER RETURNS
7931                                    ;     FOLLOWING AN INTERRUPT.
7932                                    ;
7933                                    ; ON EXIT:  @DSKETTE_STATUS, CY REFLECT STATUS OF OPERATION.
7934                                    ;           AX,BX,CX,DX DESTROYED
7935                                    ;--------------------------------------------------------------------------
7936                                    RESULTS:
7937 00002B8E 57                          PUSH   eDI
7938 00002B8F BF[C61D0000]                MOV    eDI, NEC_STATUS           ; POINTER TO DATA AREA
7939 00002B94 B307                        MOV    BL,7                ; MAX STATUS BYTES
7940 00002B96 66BAF403                    MOV    DX,03F4H            ; STATUS PORT
7941
7942                                    ;-----      WAIT FOR REQUEST FOR MASTER
7943
7944                                    _R10:
7945                                          ; 16/12/2014
7946                                          ; wait for (max) 0.5 seconds
7947                                          ;MOV   BH,2                ; HIGH ORDER COUNTER
7948                                          ;XOR   CX,CX               ; COUNTER
7949
7950                                          ;Time to wait while waiting for each byte of NEC results = .5
7951                                          ;seconds.  .5 seconds = 500,000 micros.  500,000/30 = 16,667.
7952 00002B9A 66B91B41                    mov    cx, WAIT_FDU_RESULTS_LO   ; 16667
7953 00002B9E B701                        mov    bh, WAIT_FDU_RESULTS_HI+1 ; 0+1
7954
7955                                    WFPSR_OUTER_LP:
7956                                          ;
7957                                    WFPSR_CHECK_PORT:
```

```
7958                                    J39:                            ; WAIT FOR MASTER
7959 00002BA0 EC                                IN      AL,DX           ; GET STATUS
7960 00002BA1 24C0                              AND     AL,11000000B    ; KEEP ONLY STATUS AND DIRECTION
7961 00002BA3 3CC0                              CMP     AL,11000000B    ; STATUS 1 AND DIRECTION 1 ?
7962 00002BA5 741C                              JZ      short J42       ; STATUS AND DIRECTION OK
7963                                    WFPSR_HI:
7964 00002BA7 E461                              IN      AL, PORT_B   ;061h  ; SYS1 ; wait for hi to lo
7965 00002BA9 A810                              TEST    AL,010H             ; transition on memory
7966 00002BAB 75FA                              JNZ     SHORT WFPSR_HI      ; refresh.
7967                                    WFPSR_LO:
7968 00002BAD E461                              IN      AL, PORT_B      ; SYS1
7969 00002BAF A810                              TEST    AL,010H
7970 00002BB1 74FA                              JZ      SHORT WFPSR_LO
7971 00002BB3 E2EB                                LOOP    WFPSR_CHECK_PORT
7972                                            ;
7973 00002BB5 FECF                              dec     bh
7974 00002BB7 75E7                              jnz     short WFPSR_OUTER_LP
7975                                            ;jmp    short WFPSR_TIMEOUT ; fail
7976
7977                                            ;;mov  byte [wait_count], 0
7978                                    ;J39:                           ; WAIT FOR MASTER
7979                                    ;       IN      AL,DX           ; GET STATUS
7980                                    ;       AND     AL,11000000B    ; KEEP ONLY STATUS AND DIRECTION
7981                                    ;       CMP     AL,11000000B    ; STATUS 1 AND DIRECTION 1 ?
7982                                    ;       JZ      short J42       ; STATUS AND DIRECTION OK
7983                                            ;LOOP   J39             ; LOOP TILL TIMEOUT
7984                                            ;DEC    BH              ; DECREMENT HIGH ORDER COUNTER
7985                                            ;JNZ    short J39       ; REPEAT TILL DELAY DONE
7986                                            ;
7987                                            ;;cmp  byte [wait_count], 10  ; (10/18.2 seconds)
7988                                            ;;jb   short J39
7989
7990                                    ;WFPSR_TIMEOUT:
7991 00002BB9 800D[C51D0000]80                  OR      byte [DSKETTE_STATUS],TIME_OUT
7992 00002BC0 F9                                STC                     ; SET ERROR RETURN
7993 00002BC1 EB27                              JMP     SHORT POPRES    ; POP REGISTERS AND RETURN
7994
7995                                    ;-----      READ IN THE STATUS
7996
7997                                    J42:
7998                                            ;JMP    $+2             ; I/O DELAY
7999 00002BC3 6642                              INC     DX              ; POINT AT DATA PORT
8000 00002BC5 EC                                IN      AL,DX           ; GET THE DATA
8001                                            ; 16/12/2014
8002                                            NEWIODELAY
8003 00002BC6 E6EB                  <1>  out 0ebh,al
8004 00002BC8 8807                                MOV     [eDI],AL            ; STORE THE BYTE
8005 00002BCA 47                                INC     eDI             ; INCREMENT THE POINTER
8006                                            ; 16/12/2014
```

```
8007                                    ;     push   cx
8008                                    ;     mov    cx, 30
8009                                    ;wdw2:
8010                                    ;     NEWIODELAY
8011                                    ;     loop   wdw2
8012                                    ;     pop    cx
8013
8014 00002BCB B903000000                        MOV    eCX,3              ; MINIMUM 24 MICROSECONDS FOR NEC
8015 00002BD0 E880EEFFFF                        CALL   WAITF              ; WAIT 30 TO 45 MICROSECONDS
8016 00002BD5 664A                              DEC    DX                 ; POINT AT STATUS PORT
8017 00002BD7 EC                                IN     AL,DX              ; GET STATUS
8018                                            ; 16/12/2014
8019                                            NEWIODELAY
8020 00002BD8 E6EB                      <1>  out 0ebh,al
8021                                            ;
8022 00002BDA A810                              TEST   AL,00010000B       ; TEST FOR NEC STILL BUSY
8023 00002BDC 740C                              JZ     short POPRES       ; RESULTS DONE ?
8024
8025 00002BDE FECB                              DEC    BL                 ; DECREMENT THE STATUS COUNTER
8026 00002BE0 75B8                                JNZ    short _R10              ; GO BACK FOR MORE
8027 00002BE2 800D[C51D0000]20                  OR     byte [DSKETTE_STATUS],BAD_NEC ; TOO MANY STATUS BYTES
8028 00002BE9 F9                                STC                       ; SET ERROR FLAG
8029
8030                                    ;-----      RESULT OPERATION IS DONE
8031                                    POPRES:
8032 00002BEA 5F                                POP    eDI
8033 00002BEB C3                                RETn                      ; RETURN WITH CARRY SET
8034
8035                                    ;--------------------------------------------------------------------------------
8036                                    ; READ_DSKCHNG
8037                                    ;     READS THE STATE OF THE DISK CHANGE LINE.
8038                                    ;
8039                                    ; ON ENTRY:  DI = DRIVE #
8040                                    ;
8041                                    ; ON EXIT:   DI = DRIVE #
8042                                    ;            ZF = 0 : DISK CHANGE LINE INACTIVE
8043                                    ;            ZF = 1 : DISK CHANGE LINE ACTIVE
8044                                    ;            AX,CX,DX DESTROYED
8045                                    ;--------------------------------------------------------------------------------
8046                                    READ_DSKCHNG:
8047 00002BEC E8ADFDFFFF                        CALL   MOTOR_ON           ; TURN ON THE MOTOR IF OFF
8048 00002BF1 66BAF703                          MOV    DX,03F7H           ; ADDRESS DIGITAL INPUT REGISTER
8049 00002BF5 EC                                IN     AL,DX              ; INPUT DIGITAL INPUT REGISTER
8050 00002BF6 A880                              TEST   AL,DSK_CHG         ; CHECK FOR DISK CHANGE LINE ACTIVE
8051 00002BF8 C3                                RETn                      ; RETURN TO CALLER WITH ZERO FLAG SET
8052
8053                                    ;--------------------------------------------------------------------------------
8054                                    ; DRIVE_DET
8055                                    ;     DETERMINES WHETHER DRIVE IS 80 OR 40 TRACKS AND
```

```
8056                                    ;     UPDATES STATE INFORMATION ACCORDINGLY.
8057                                    ;³-pol*-j    56789*87111"; ON ENTRY:   DI = DRIVE #
8058                                    ;-------------------------------------------------------------------------------
8059                                    DRIVE_DET:
8060 00002BF9 E8A0FDFFFF                      CALL   MOTOR_ON            ; TURN ON MOTOR IF NOT ALREADY ON
8061 00002BFE E80AFFFFFF                      CALL   RECAL               ; RECALIBRATE DRIVE
8062 00002C03 7251                            JC     short DD_BAC        ; ASSUME NO DRIVE PRESENT
8063 00002C05 B530                            MOV    CH,TRK_SLAP         ; SEEK TO TRACK 48
8064 00002C07 E882FEFFFF                      CALL   SEEK
8065 00002C0C 7248                            JC     short DD_BAC        ; ERROR NO DRIVE
8066 00002C0E B50B                            MOV    CH,QUIET_SEEK+1           ; SEEK TO TRACK 10
8067                                    SK_GIN:
8068 00002C10 FECD                            DEC    CH                  ; DECREMENT TO NEXT TRACK
8069 00002C12 6651                            PUSH   CX                  ; SAVE TRACK
8070 00002C14 E875FEFFFF                      CALL   SEEK
8071 00002C19 723C                            JC     short POP_BAC       ; POP AND RETURN
8072 00002C1B B8[572C0000]                    MOV    eAX, POP_BAC        ; LOAD NEC OUTPUT ERROR ADDRESS
8073 00002C20 50                              PUSH   eAX
8074 00002C21 B404                            MOV    AH,SENSE_DRV_ST           ; SENSE DRIVE STATUS COMMAND BYTE
8075 00002C23 E837FEFFFF                      CALL   NEC_OUTPUT          ; OUTPUT TO NEC
8076 00002C28 6689F8                          MOV    AX,DI               ; AL = DRIVE
8077 00002C2B 88C4                            MOV    AH,AL               ; AH = DRIVE
8078 00002C2D E82DFEFFFF                      CALL   NEC_OUTPUT          ; OUTPUT TO NEC
8079 00002C32 E857FFFFFF                      CALL   RESULTS                   ; GO GET STATUS
8080 00002C37 58                              POP    eAX                 ; THROW AWAY ERROR ADDRESS
8081 00002C38 6659                            POP    CX                  ; RESTORE TRACK
8082 00002C3A F605[C61D0000]10                TEST   byte [NEC_STATUS], HOME   ; TRACK 0 ?
8083 00002C41 74CD                            JZ     short SK_GIN        ; GO TILL TRACK 0
8084 00002C43 08ED                            OR     CH,CH               ; IS HOME AT TRACK 0
8085 00002C45 7408                            JZ     short IS_80         ; MUST BE 80 TRACK DRIVE
8086
8087                                    ;     DRIVE IS A 360; SET DRIVE TO DETERMINED;
8088                                    ;     SET MEDIA TO DETERMINED AT RATE 250.
8089
8090 00002C47 808F[D31D0000]94                OR     byte [DSK_STATE+eDI], DRV_DET+MED_DET+RATE_250
8091 00002C4E C3                              RETn                       ; ALL INFORMATION SET
8092                                    IS_80:
8093 00002C4F 808F[D31D0000]01                OR     byte [DSK_STATE+eDI], TRK_CAPA ; SETUP 80 TRACK CAPABILITY
8094                                    DD_BAC:
8095 00002C56 C3                              RETn
8096                                    POP_BAC:
8097 00002C57 6659                            POP    CX                  ; THROW AWAY
8098 00002C59 C3                              RETn
8099
8100                                    fdc_int:  ; 16/02/2015
8101                                    ;int_0Eh: ; 11/12/2014
8102
8103                                    ;--- HARDWARE INT 0EH -- ( IRQ LEVEL  6 ) -------------------------------------
8104                                    ; DISK_INT
```

```
8105                                    ;     THIS ROUTINE HANDLES THE DISKETTE INTERRUPT.
8106                                    ;
8107                                    ; ON EXIT:  THE INTERRUPT FLAG IS SET IN @SEEK_STATUS.
8108                                    ;--------------------------------------------------------------------------------
8109                                    DISK_INT_1:
8110
8111 00002C5A 6650                           PUSH   AX                 ; SAVE WORK REGISTER
8112                                          ;OR      byte [CS:SEEK_STATUS], INT_FLAG ; TURN ON INTERRUPT OCCURRED
8113 00002C5C 800D[C21D0000]80               or   byte [SEEK_STATUS], INT_FLAG ; 08/02/2015
8114 00002C63 B020                           MOV     AL,EOI                ; END OF INTERRUPT MARKER
8115 00002C65 E620                           OUT    INTA00,AL          ; INTERRUPT CONTROL PORT
8116 00002C67 6658                           POP    AX                 ; RECOVER REGISTER
8117 00002C69 CF                             IRET                      ; RETURN FROM INTERRUPT
8118
8119                                    ;--------------------------------------------------------------------------------
8120                                    ; DSKETTE_SETUP
8121                                    ;     THIS ROUTINE DOES A PRELIMINARY CHECK TO SEE WHAT TYPE OF
8122                                    ;     DISKETTE DRIVES ARE ATTACH TO THE SYSTEM.
8123                                    ;--------------------------------------------------------------------------------
8124                                    DSKETTE_SETUP:
8125                                          ;PUSH  AX                 ; SAVE REGISTERS
8126                                          ;PUSH  BX
8127                                          ;PUSH  CX
8128 00002C6A 52                             PUSH   eDX
8129                                          ;PUSH  DI
8130                                          ;;PUSH DS
8131                                          ; 14/12/2014
8132                                          ;mov   word [DISK_POINTER], MD_TBL6
8133                                          ;mov   word [DISK_POINTER+2], cs
8134                                          ;
8135                                          ;OR    byte [RTC_WAIT_FLAG], 1   ; NO RTC WAIT, FORCE USE OF LOOP
8136 00002C6B 31FF                           XOR    eDI,eDI                   ; INITIALIZE DRIVE POINTER
8137 00002C6D 66C705[D31D0000]00-            MOV    WORD [DSK_STATE],0  ; INITIALIZE STATES
8138 00002C75 00
8139 00002C76 8025[D21D0000]33               AND    byte [LASTRATE],~(STRT_MSK+SEND_MSK) ; CLEAR START & SEND
8140 00002C7D 800D[D21D0000]C0               OR     byte [LASTRATE],SEND_MSK ; INITIALIZE SENT TO IMPOSSIBLE
8141 00002C84 C605[C21D0000]00               MOV    byte [SEEK_STATUS],0      ; INDICATE RECALIBRATE NEEDED
8142 00002C8B C605[C41D0000]00               MOV    byte [MOTOR_COUNT],0      ; INITIALIZE MOTOR COUNT
8143 00002C92 C605[C31D0000]00               MOV    byte [MOTOR_STATUS],0     ; INITIALIZE DRIVES TO OFF STATE
8144 00002C99 C605[C51D0000]00               MOV    byte [DSKETTE_STATUS],0   ; NO ERRORS
8145
8146                                          ; 28/02/2015
8147                                          ;mov   word [cfd], 100h
8148 00002CA0 E87AF2FFFF                     call   DSK_RESET
8149 00002CA5 5A                             pop    edx
8150 00002CA6 C3                             retn
8151
8152                                    ;SUP0:
8153                                    ;     CALL   DRIVE_DET          ; DETERMINE DRIVE
```

```
8154                                    ;     CALL   XLAT_OLD           ; TRANSLATE STATE TO COMPATIBLE MODE
8155                                    ;     ; 02/01/2015
8156                                    ;     ;INC   DI                 ; POINT TO NEXT DRIVE
8157                                    ;     ;CMP   DI,MAX_DRV         ; SEE IF DONE
8158                                    ;     ;JNZ   short SUP0         ; REPEAT FOR EACH ORIVE
8159                                    ;       cmp     byte [fd1_type], 0
8160                                    ;     jna    short sup1
8161                                    ;     or     di, di
8162                                    ;     jnz    short sup1
8163                                    ;     inc    di
8164                                    ;       jmp     short SUP0
8165                                    ;sup1:
8166                                    ;     MOV    byte [SEEK_STATUS],0      ; FORCE RECALIBRATE
8167                                    ;     ;AND   byte [RTC_WAIT_FLAG],0FEH ; ALLOW FOR RTC WAIT
8168                                    ;     CALL   SETUP_END          ; VARIOUS CLEANUPS
8169                                    ;     ;;POP  DS                 ; RESTORE CALLERS REGISTERS
8170                                    ;     ;POP   DI
8171                                    ;     POP    eDX
8172                                    ;     ;POP   CX
8173                                    ;     ;POP   BX
8174                                    ;     ;POP   AX
8175                                    ;     RETn
8176
8177                                    ;////////////////////////////////////////////////
8178                                    ;; END OF DISKETTE I/O ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
8179                                    ;
8180
8181                                    int13h: ; 21/02/2015
8182 00002CA7 9C                             pushfd
8183 00002CA8 0E                             push   cs
8184 00002CA9 E875000000                     call   DISK_IO
8185 00002CAE C3                             retn
8186
8187                                    ;;;;;; DISK I/O ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; 21/02/2015 ;;;
8188                                    ;//////////////////////////////////////////////////////////////////////
8189
8190                                    ; DISK I/O - Erdogan Tan (Retro UNIX 386 v1 project)
8191                                    ; 23/02/2015
8192                                    ; 21/02/2015 (unix386.s)
8193                                    ; 22/12/2014 - 14/02/2015 (dsectrm2.s)
8194                                    ;
8195                                    ; Original Source Code:
8196                                    ; DISK ----- 09/25/85 FIXED DISK BIOS
8197                                    ; (IBM PC XT Model 286 System BIOS Source Code, 04-21-86)
8198                                    ;
8199                                    ; Modifications: by reference of AWARD BIOS 1999 (D1A0622)
8200                                    ;            Source Code - ATORGS.ASM, AHDSK.ASM
8201                                    ;
8202
```

```
8203
8204                                    ;The wait for controller to be not busy is 10 seconds.
8205                                    ;10,000,000 / 30 = 333,333.  333,333 decimal = 051615h
8206                                    ;;WAIT_HDU_CTLR_BUSY_LO   equ    1615h
8207                                    ;;WAIT_HDU_CTLR_BUSY_HI   equ      05h
8208                                    WAIT_HDU_CTRL_BUSY_LH     equ    51615h  ;21/02/2015
8209
8210                                    ;The wait for controller to issue completion interrupt is 10 seconds.
8211                                    ;10,000,000 / 30 = 333,333.  333,333 decimal = 051615h
8212                                    ;;WAIT_HDU_INT_LO  equ    1615h
8213                                    ;;WAIT_HDU_INT_HI  equ      05h
8214                                    WAIT_HDU_INT_LH           equ    51615h ; 21/02/2015
8215
8216                                    ;The wait for Data request on read and write longs is
8217                                    ;2000 us. (?)
8218                                    ;;WAIT_HDU_DRQ_LO  equ    1000   ; 03E8h
8219                                    ;;WAIT_HDU_DRQ_HI  equ    0
8220                                    WAIT_HDU_DRQ_LH           equ    1000   ; 21/02/2015
8221
8222                                    ; Port 61h (PORT_B)
8223                                    SYS1        equ    61h    ; PORT_B  (diskette.inc)
8224
8225                                    ; 23/12/2014
8226                                    %define CMD_BLOCK       eBP-8  ; 21/02/2015
8227
8228
8229                                    ; << diskette.inc >>
8230                                    ; ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
8231                                    ;
8232                                    ;---------------------------------------
8233                                    ;     ROM BIOS DATA AREAS       :
8234                                    ;---------------------------------------
8235
8236                                    ;DATA       SEGMENT AT 40H            ; ADDRESS= 0040:0000
8237
8238                                    ;---------------------------------------
8239                                    ;     FIXED DISK DATA AREAS           :
8240                                    ;---------------------------------------
8241
8242 00002CAF 00                        DISK_STATUS1:      DB     0            ; FIXED DISK STATUS
8243 00002CB0 00                        HF_NUM:            DB     0            ; COUNT OF FIXED DISK DRIVES
8244 00002CB1 00                        CONTROL_BYTE:      DB     0            ; HEAD CONTROL BYTE
8245                                    ;@PORT_OFF   DB    ?            ;  RESERVED (PORT OFFSET)
8246
8247                                    ;---------------------------------------
8248                                    ;     ADDITIONAL MEDIA DATA           :
8249                                    ;---------------------------------------
8250
8251                                    ;@LASTRATE   DB    ?            ; LAST DISKETTE DATA RATE SELECTED
```

```
8252 00002CB2 00                HF_STATUS   DB    0           ; STATUS REGISTER
8253 00002CB3 00                HF_ERROR    DB    0           ; ERROR REGISTER
8254 00002CB4 00                HF_INT_FLAG DB    0           ; FIXED DISK INTERRUPT FLAG
8255 00002CB5 00                HF_CNTRL    DB    0           ; COMBO FIXED DISK/DISKETTE CARD BIT 0=1
8256                            ;@DSK_STATE DB    ?           ; DRIVE 0 MEDIA STATE
8257                            ;           DB    ?           ; DRIVE 1 MEDIA STATE
8258                            ;           DB    ?           ; DRIVE 0 OPERATION START STATE
8259                            ;           DB    ?           ; DRIVE 1 OPERATION START STATE
8260                            ;@DSK_TRK   DB    ?           ; DRIVE 0 PRESENT CYLINDER
8261                            ;           DB    ?           ; DRIVE 1 PRESENT CYLINDER
8262
8263                            ;DATA       ENDS              ; END OF BIOS DATA SEGMENT
8264                            ;
8265                            ; ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
8266
8267
8268                            ;--- INT 13H ----------------------------------------------------------------------
8269                            ;                                                                              :
8270                            ; FIXED DISK I/O INTERFACE                                                     :
8271                            ;                                                                              :
8272                            ;    THIS INTERFACE PROVIDES ACCESS TO 5 1/4" FIXED DISKS THROUGH              :
8273                            ;    THE IBM FIXED DISK CONTROLLER.                                            :
8274                            ;                                                                              :
8275                            ;    THE  BIOS  ROUTINES  ARE  MEANT  TO  BE  ACCESSED  THROUGH               :
8276                            ;    SOFTWARE  INTERRUPTS  ONLY.    ANY  ADDRESSES  PRESENT     IN             :
8277                            ;    THESE  LISTINGS  ARE  INCLUDED    ONLY FOR  COMPLETENESS,                :
8278                            ;    NOT  FOR  REFERENCE.  APPLICATIONS   WHICH  REFERENCE  ANY               :
8279                            ;    ABSOLUTE  ADDRESSES  WITHIN  THE  CODE SEGMENTS  OF  BIOS                :
8280                            ;    VIOLATE  THE  STRUCTURE  AND  DESIGN  OF  BIOS.                           :
8281                            ;                                                                              :
8282                            ;------------------------------------------------------------------------------:
8283                            ;                                                                              :
8284                            ; INPUT  (AH)= HEX COMMAND VALUE                                               :
8285                            ;                                                                              :
8286                            ;    (AH)= 00H  RESET DISK (DL = 80H,81H) / DISKETTE                           :
8287                            ;    (AH)= 01H  READ THE STATUS OF THE LAST DISK OPERATION INTO (AL)           :
8288                            ;               NOTE: DL < 80H - DISKETTE                                      :
8289                            ;                     DL > 80H - DISK                                          :
8290                            ;    (AH)= 02H  READ THE DESIRED SECTORS INTO MEMORY                           :
8291                            ;    (AH)= 03H  WRITE THE DESIRED SECTORS FROM MEMORY                          :
8292                            ;    (AH)= 04H  VERIFY THE DESIRED SECTORS                                     :
8293                            ;    (AH)= 05H  FORMAT THE DESIRED TRACK                                       :
8294                            ;    (AH)= 06H  UNUSED                                                         :
8295                            ;    (AH)= 07H  UNUSED                                                         :
8296                            ;    (AH)= 08H  RETURN THE CURRENT DRIVE PARAMETERS                            :
8297                            ;    (AH)= 09H  INITIALIZE DRIVE PAIR CHARACTERISTICS                          :
8298                            ;               INTERRUPT 41 POINTS TO DATA BLOCK FOR DRIVE 0                  :
8299                            ;               INTERRUPT 46 POINTS TO DATA BLOCK FOR DRIVE 1                  :
8300                            ;    (AH)= 0AH  READ LONG                                                      :
```

```
;     (AH)= 0BH  WRITE LONG  (READ & WRITE LONG ENCOMPASS 512 + 4 BYTES ECC) :
;     (AH)= 0CH  SEEK                                                       :
;     (AH)= 0DH  ALTERNATE DISK RESET (SEE DL)                                :
;     (AH)= 0EH  UNUSED                                                     :
;     (AH)= 0FH  UNUSED                                                     :
;     (AH)= 10H  TEST DRIVE READY                                             :
;     (AH)= 11H  RECALIBRATE                                                  :
;     (AH)= 12H  UNUSED                                                     :
;     (AH)= 13H  UNUSED                                                     :
;     (AH)= 14H  CONTROLLER INTERNAL DIAGNOSTIC                               :
;     (AH)= 15H  READ DASD TYPE                                             :
;                                                                           :
;---------------------------------------------------------------------------
;                                                                           :
;     REGISTERS USED FOR FIXED DISK OPERATIONS                                :
;                                                                           :
;           (DL)   -  DRIVE NUMBER     (80H-81H FOR DISK. VALUE CHECKED)  :
;           (DH)   -  HEAD NUMBER          (0-15 ALLOWED, NOT VALUE CHECKED)  :
;           (CH)   -  CYLINDER NUMBER  (0-1023, NOT VALUE CHECKED)(SEE CL):
;           (CL)   -  SECTOR NUMBER    (1-17, NOT VALUE CHECKED)        :
;                                                                           :
;                     NOTE: HIGH 2 BITS OF CYLINDER NUMBER ARE PLACED     :
;                          IN THE HIGH 2 BITS OF THE CL REGISTER          :
;                          (10 BITS TOTAL)                                :
;                                                                           :
;           (AL)   -  NUMBER OF SECTORS (MAXIMUM POSSIBLE RANGE 1-80H,    :
;                                        FOR READ/WRITE LONG 1-79H)       :
;                                                                           :
;           (ES:BX) -  ADDRESS OF BUFFER FOR READS AND WRITES,              :
;                     (NOT REQUIRED FOR VERIFY)                             :
;                                                                           :
;           FORMAT (AH=5) ES:BX POINTS TO A 512 BYTE BUFFER. THE FIRST     :
;                     2*(SECTORS/TRACK) BYTES CONTAIN F,N FOR EACH SECTOR.:
;                     F = 00H FOR A GOOD SECTOR                             :
;                         80H FOR A BAD SECTOR                              :
;                     N = SECTOR NUMBER                                     :
;                     FOR AN INTERLEAVE OF 2 AND 17 SECTORS/TRACK             :
;                     THE TABLE SHOULD BE:                                  :
;                                                                           :
;              DB     00H,01H,00H,0AH,00H,02H,00H,0BH,00H,03H,00H,0CH     :
;              DB     00H,04H,00H,0DH,00H,05H,00H,0EH,00H,06H,00H,0FH     :
;              DB     00H,07H,00H,10H,00H,08H,00H,11H,00H,09H             :
;                                                                           :
;---------------------------------------------------------------------------

;---------------------------------------------------------------------------
; OUTPUT                                                                    :
;     AH = STATUS OF CURRENT OPERATION                                      :
;          STATUS BITS ARE DEFINED IN THE EQUATES BELOW                       :
```

```
8350                    ;     CY = 0 SUCCESSFUL OPERATION (AH=0 ON RETURN)                    :
8351                    ;     CY = 1 FAILED OPERATION (AH HAS ERROR REASON)                   :
8352                    ;                                                                     :
8353                    ;     NOTE:  ERROR 11H  INDICATES THAT THE DATA READ HAD A RECOVERABLE      :
8354                    ;            ERROR WHICH WAS CORRECTED BY THE ECC ALGORITHM.  THE DATA      :
8355                    ;            IS PROBABLY GOOD,   HOWEVER THE BIOS ROUTINE INDICATES AN      :
8356                    ;            ERROR TO ALLOW THE CONTROLLING PROGRAM A CHANCE TO DECIDE      :
8357                    ;            FOR ITSELF.  THE  ERROR  MAY  NOT  RECUR  IF  THE DATA IS     :
8358                    ;            REWRITTEN.                                                    :
8359                    ;                                                                          :
8360                    ;     IF DRIVE PARAMETERS WERE REQUESTED (DL >= 80H),                 :
8361                    ;        INPUT:                                                            :
8362                    ;          (DL) = DRIVE NUMBER                                             :
8363                    ;        OUTPUT:                                                           :
8364                    ;          (DL) = NUMBER OF CONSECUTIVE ACKNOWLEDGING DRIVES ATTACHED (1-2)  :
8365                    ;               (CONTROLLER CARD ZERO TALLY ONLY)                     :
8366                    ;          (DH) = MAXIMUM USEABLE VALUE FOR HEAD NUMBER                  :
8367                    ;          (CH) = MAXIMUM USEABLE VALUE FOR CYLINDER NUMBER                    :
8368                    ;          (CL) = MAXIMUM USEABLE VALUE FOR SECTOR NUMBER                :
8369                    ;               AND CYLINDER NUMBER HIGH BITS                            :
8370                    ;                                                                          :
8371                    ;     IF READ DASD TYPE WAS REQUESTED,                                     :
8372                    ;                                                                          :
8373                    ;     AH = 0 - NOT PRESENT                                                 :
8374                    ;          1 - DISKETTE - NO CHANGE LINE AVAILABLE                         :
8375                    ;          2 - DISKETTE - CHANGE LINE AVAILABLE                            :
8376                    ;          3 - FIXED DISK                                                  :
8377                    ;                                                                          :
8378                    ;     CX,DX = NUMBER OF 512 BYTE BLOCKS WHEN AH = 3                        :
8379                    ;                                                                          :
8380                    ;     REGISTERS WILL BE PRESERVED EXCEPT WHEN THEY ARE USED TO RETURN         :
8381                    ;     INFORMATION.                                                         :
8382                    ;                                                                          :
8383                    ;     NOTE: IF AN ERROR IS REPORTED BY THE DISK CODE, THE APPROPRIATE         :
8384                    ;           ACTION IS TO RESET THE DISK, THEN RETRY THE OPERATION.             :
8385                    ;                                                                          :
8386                    ;-------------------------------------------------------------------------------
8387
8388                    SENSE_FAIL   EQU    0FFH          ; NOT IMPLEMENTED
8389                    NO_ERR       EQU    0E0H          ; STATUS ERROR/ERROR REGISTER=0
8390                    WRITE_FAULT  EQU    0CCH          ; WRITE FAULT ON SELECTED DRIVE
8391                    UNDEF_ERR    EQU    0BBH          ; UNDEFINED ERROR OCCURRED
8392                    NOT_RDY      EQU    0AAH          ; DRIVE NOT READY
8393                    TIME_OUT     EQU    80H           ; ATTACHMENT FAILED TO RESPOND
8394                    BAD_SEEK     EQU    40H           ; SEEK OPERATION FAILED
8395                    BAD_CNTLR    EQU    20H           ; CONTROLLER HAS FAILED
8396                    DATA_CORRECTED     EQU    11H          ; ECC CORRECTED DATA ERROR
8397                    BAD_ECC      EQU    10H           ; BAD ECC ON DISK READ
8398                    BAD_TRACK    EQU    0BH           ; NOT IMPLEMENTED
```

```
BAD_SECTOR   EQU    0AH          ; BAD SECTOR FLAG DETECTED
;DMA_BOUNDARY      EQU    09H          ; DATA EXTENDS TOO FAR
INIT_FAIL    EQU    07H          ; DRIVE PARAMETER ACTIVITY FAILED
BAD_RESET    EQU    05H          ; RESET FAILED
;RECORD_NOT_FND    EQU    04H          ; REQUESTED SECTOR NOT FOUND
;BAD_ADDR_MARK     EQU    02H          ; ADDRESS MARK NOT FOUND
;BAD_CMD     EQU    01H          ; BAD COMMAND PASSED TO DISK I/O

;--------------------------------------------------------
;                                                       :
; FIXED DISK PARAMETER TABLE                            :
;  -  THE TABLE IS COMPOSED OF A BLOCK DEFINED AS:      :
;                                                       :
;  +0 (1 WORD) - MAXIMUM NUMBER OF CYLINDERS            :
;  +2 (1 BYTE) - MAXIMUM NUMBER OF HEADS                :
;  +3 (1 WORD) - NOT USED/SEE PC-XT                     :
;  +5 (1 WORD) - STARTING WRITE PRECOMPENSATION CYL     :
;  +7 (1 BYTE) - MAXIMUM ECC DATA BURST LENGTH          :
;  +8 (1 BYTE) - CONTROL BYTE                           :
;              BIT   7 DISABLE RETRIES -OR-             :
;              BIT   6 DISABLE RETRIES                  :
;              BIT   3 MORE THAN 8 HEADS                :
;  +9 (3 BYTES)- NOT USED/SEE PC-XT                     :
; +12 (1 WORD) - LANDING ZONE                           :
; +14 (1 BYTE) - NUMBER OF SECTORS/TRACK                :
; +15 (1 BYTE) - RESERVED FOR FUTURE USE                :
;                                                       :
;      - TO DYNAMICALLY DEFINE A SET OF PARAMETERS      :
;        BUILD A TABLE FOR UP TO 15 TYPES AND PLACE     :
;        THE CORRESPONDING VECTOR INTO INTERRUPT 41     :
;        FOR DRIVE 0 AND INTERRUPT 46 FOR DRIVE 1.      :
;                                                       :
;--------------------------------------------------------

;--------------------------------------------------------
;                                                       :
; HARDWARE SPECIFIC VALUES                              :
;                                                       :
;  -  CONTROLLER I/O PORT                               :
;                                                       :
;     > WHEN READ FROM:                                 :
;     HF_PORT+0 - READ DATA (FROM CONTROLLER TO CPU)          :
;     HF_PORT+1 - GET ERROR REGISTER                    :
;     HF_PORT+2 - GET SECTOR COUNT                      :
;     HF_PORT+3 - GET SECTOR NUMBER                     :
;     HF_PORT+4 - GET CYLINDER LOW                      :
;     HF_PORT+5 - GET CYLINDER HIGH (2 BITS)            :
;     HF_PORT+6 - GET SIZE/DRIVE/HEAD                   :
;     HF_PORT+7 - GET STATUS REGISTER                   :
```

```
8448                                    ;                                               :
8449                                    ;     > WHEN WRITTEN TO:                        :
8450                                    ;     HF_PORT+0 - WRITE DATA (FROM CPU TO CONTROLLER) :
8451                                    ;     HF_PORT+1 - SET PRECOMPENSATION CYLINDER      :
8452                                    ;     HF_PORT+2 - SET SECTOR COUNT                  :
8453                                    ;     HF_PORT+3 - SET SECTOR NUMBER                 :
8454                                    ;     HF_PORT+4 - SET CYLINDER LOW                  :
8455                                    ;     HF_PORT+5 - SET CYLINDER HIGH (2 BITS)        :
8456                                    ;     HF_PORT+6 - SET SIZE/DRIVE/HEAD         :
8457                                    ;     HF_PORT+7 - SET COMMAND REGISTER        :
8458                                    ;                                             :
8459                                    ;-----------------------------------------------------
8460
8461                                    ;HF_PORT     EQU    01F0H  ; DISK PORT
8462                                    ;HF1_PORT    equ    0170h
8463                                    ;HF_REG_PORT EQU    03F6H
8464                                    ;HF1_REG_PORT       equ    0376h
8465
8466                                    align 2
8467
8468 00002CB6 F001                     HF_PORT:     dw     1F0h  ; Default = 1F0h
8469                                                         ; (170h)
8470 00002CB8 F603                     HF_REG_PORT: dw     3F6h  ; HF_PORT + 206h
8471
8472                                    HDC1_BASEPORT       equ    1F0h
8473                                    HDC2_BASEPORT       equ    170h
8474
8475                                    ; 05/01/2015
8476 00002CBA 00                       hf_m_s:         db      0     ; (0 = Master, 1 = Slave)
8477
8478 00002CBB 90                       align 2
8479
8480                                    ;-----              STATUS REGISTER
8481
8482                                    ST_ERROR     EQU    00000001B    ;
8483                                    ST_INDEX     EQU    00000010B    ;
8484                                    ST_CORRCTD   EQU    00000100B    ; ECC CORRECTION SUCCESSFUL
8485                                    ST_DRQ              EQU    00001000B    ;
8486                                    ST_SEEK_COMPL       EQU    00010000B    ; SEEK COMPLETE
8487                                    ST_WRT_FLT   EQU    00100000B    ; WRITE FAULT
8488                                    ST_READY     EQU    01000000B    ;
8489                                    ST_BUSY      EQU    10000000B    ;
8490
8491                                    ;-----              ERROR REGISTER
8492
8493                                    ERR_DAM      EQU    00000001B    ; DATA ADDRESS MARK NOT FOUND
8494                                    ERR_TRK_0    EQU    00000010B    ; TRACK 0 NOT FOUND ON RECAL
8495                                    ERR_ABORT    EQU    00000100B    ; ABORTED COMMAND
8496                                    ;            EQU    00001000B    ; NOT USED
```

```
                                ERR_ID              EQU    00010000B    ; ID NOT FOUND
                                ;           EQU     00100000B    ; NOT USED
                                ERR_DATA_ECC EQU    01000000B
                                ERR_BAD_BLOCK       EQU    10000000B


                                RECAL_CMD    EQU    00010000B    ; DRIVE RECAL(10H)
                                READ_CMD     EQU    00100000B    ;      READ   (20H)
                                WRITE_CMD    EQU    00110000B    ;      WRITE  (30H)
                                VERIFY_CMD   EQU    01000000B    ;      VERIFY (40H)
                                FMTTRK_CMD   EQU    01010000B    ; FORMAT TRACK      (50H)
                                INIT_CMD     EQU    01100000B    ;   INITIALIZE      (60H)
                                SEEK_CMD     EQU    01110000B    ;      SEEK   (70H)
                                DIAG_CMD     EQU    10010000B    ; DIAGNOSTIC (90H)
                                SET_PARM_CMD EQU    10010001B    ; DRIVE PARMS(91H)
                                NO_RETRIES   EQU    00000001B    ; CHD MODIFIER      (01H)
                                ECC_MODE     EQU    00000010B    ; CMD MODIFIER      (02H)
                                BUFFER_MODE  EQU    00001000B    ; CMD MODIFIER      (08H)

                                ;MAX_FILE    EQU    2
                                ;S_MAX_FILE  EQU    2
                                MAX_FILE     equ    4            ; 22/12/2014
                                S_MAX_FILE   equ    4            ; 22/12/2014

                                DELAY_1      EQU    25H          ; DELAY FOR OPERATION COMPLETE
                                DELAY_2      EQU    0600H        ; DELAY FOR READY
                                DELAY_3      EQU    0100H        ; DELAY FOR DATA REQUEST

                                HF_FAIL      EQU    08H          ; CMOS FLAG IN BYTE 0EH

                                ;-----              COMMAND BLOCK REFERENCE

                                ;CMD_BLOCK      EQU     BP-8            ; @CMD_BLOCK REFERENCES BLOCK HEAD IN SS
                                                                    ;  (BP) POINTS TO COMMAND BLOCK TAIL
                                                                    ;     AS DEFINED BY THE "ENTER" PARMS
                                ; 19/12/2014
                                ORG_VECTOR   equ    4*13h        ; INT 13h vector
                                DISK_VECTOR  equ    4*40h        ; INT 40h vector (for floppy disks)
                                ;HDISK_INT   equ    4*76h        ; Primary HDC - Hardware interrupt (IRQ14)
                                ;HDISK_INT1  equ    4*76h        ; Primary HDC - Hardware interrupt (IRQ14)
                                ;HDISK_INT2  equ    4*77h        ; Secondary HDC - Hardware interrupt (IRQ15)
                                ;HF_TBL_VEC  equ    4*41h        ; Pointer to 1st fixed disk parameter table
                                ;HF1_TBL_VEC equ    4*46h        ; Pointer to 2nd fixed disk parameter table
                                ;
                                ;HF_TBL_VEC: dd     0            ; Primary master disk param. tbl. pointer
                                ;HF1_TBL_VEC:       dd     0            ; Primary slave disk param. tbl. pointer
                                HF_TBL_VEC: ; 22/12/2014
00002CBC 00000000               HDPM_TBL_VEC:       dd     0            ; Primary master disk param. tbl. pointer
00002CC0 00000000               HDPS_TBL_VEC:       dd     0            ; Primary slave disk param. tbl. pointer
```

```
8546 00002CC4 00000000              HDSM_TBL_VEC:      dd    0           ; Secondary master disk param. tbl. pointer
8547 00002CC8 00000000              HDSS_TBL_VEC:      dd    0           ; Secondary slave disk param. tbl. pointer
8548
8549                                ;---------------------------------------
8550                                ;     FIXED DISK DATA AREAS           :
8551                                ;---------------------------------------
8552
8553                                ;@DISK_STATUS1     DB    ?           ; FIXED DISK STATUS
8554                                ;@HF_NUM    DB     ?           ; COUNT OF FIXED DISK DRIVES
8555                                ;@CONTROL_BYTE     DB    ?           ; HEAD CONTROL BYTE
8556                                ;@PORT_OFF  DB     ?           ;  RESERVED (PORT OFFSET)
8557                                ;
8558                                ;port1_off  db     0           ; Hard disk controller 1 - port offset
8559                                ;port2_off  db     0           ; Hard idsk controller 2 - port offset
8560
8561                                align 2
8562
8563                                ;-------------------------------------------------------------
8564                                ; FIXED DISK I/O SETUP                                       :
8565                                ;                                                  :
8566                                ;  -  ESTABLISH TRANSFER VECTORS FOR THE FIXED DISK        :
8567                                ;  -  PERFORM POWER ON DIAGNOSTICS                         :
8568                                ;     SHOULD AN ERROR OCCUR A "1701" MESSAGE IS DISPLAYED       :
8569                                ;                                                  :
8570                                ;-------------------------------------------------------------
8571
8572                                DISK_SETUP:
8573                                      ;CLI
8574                                      ;;MOV  AX,ABS0                  ; GET ABSOLUTE SEGMENT
8575                                      ;xor   ax,ax
8576                                      ;MOV   DS,AX                    ; SET SEGMENT REGISTER
8577                                      ;MOV   AX, word [ORG_VECTOR]            ; GET DISKETTE VECTOR
8578                                      ;MOV   word [DISK_VECTOR],AX            ;  INTO INT 40H
8579                                      ;MOV   AX, word [ORG_VECTOR+2]
8580                                      ;MOV   word [DISK_VECTOR+2],AX
8581                                      ;MOV   word [ORG_VECTOR],DISK_IO ; FIXED DISK HANDLER
8582                                      ;MOV   word [ORG_VECTOR+2],CS
8583                                      ; 1st controller (primary master, slave)   - IRQ 14
8584                                      ;;MOV  word [HDISK_INT],HD_INT          ; FIXED DISK INTERRUPT
8585                                      ;mov   word [HDISK_INT1],HD_INT  ;
8586                                      ;;MOV  word [HDISK_INT+2],CS
8587                                      ;mov   word [HDISK_INT1+2],CS
8588                                      ; 2nd controller (secondary master, slave) - IRQ 15
8589                                      ;mov   word [HDISK_INT2],HD1_INT ;
8590                                      ;mov   word [HDISK_INT2+2],CS
8591                                      ;
8592                                      ;;MOV  word [HF_TBL_VEC],HD0_DPT ; PARM TABLE DRIVE 80
8593                                      ;;MOV  word [HF_TBL_VEC+2],DPT_SEGM
8594                                      ;;MOV  word [HF1_TBL_VEC],HD1_DPT; PARM TABLE DRIVE 81
```

```
                                        ;;MOV  word [HF1_TBL_VEC+2],DPT_SEGM
                                        ;push  cs
                                        ;pop   ds
                                        ;mov   word [HDPM_TBL_VEC],HD0_DPT      ; PARM TABLE DRIVE 80h
                                        ;mov   word [HDPM_TBL_VEC+2],DPT_SEGM
00002CCC C705[BC2C0000]0000-            mov    dword [HDPM_TBL_VEC], (DPT_SEGM*16)+HD0_DPT
00002CD4 0900
                                        ;mov   word [HDPS_TBL_VEC],HD1_DPT      ; PARM TABLE DRIVE 81h
                                        ;mov   word [HDPS_TBL_VEC+2],DPT_SEGM
00002CD6 C705[C02C0000]2000-            mov    dword [HDPS_TBL_VEC], (DPT_SEGM*16)+HD1_DPT
00002CDE 0900
                                        ;mov   word [HDSM_TBL_VEC],HD2_DPT      ; PARM TABLE DRIVE 82h
                                        ;mov   word [HDSM_TBL_VEC+2],DPT_SEGM
00002CE0 C705[C42C0000]4000-            mov    dword [HDSM_TBL_VEC], (DPT_SEGM*16)+HD2_DPT
00002CE8 0900
                                        ;mov   word [HDSS_TBL_VEC],HD3_DPT      ; PARM TABLE DRIVE 83h
                                        ;mov   word [HDSS_TBL_VEC+2],DPT_SEGM
00002CEA C705[C82C0000]6000-            mov    dword [HDSS_TBL_VEC], (DPT_SEGM*16)+HD3_DPT
00002CF2 0900
                                        ;
                                        ;;IN   AL,INTB01          ; TURN ON SECOND INTERRUPT CHIP
                                        ;;;AND AL,0BFH
                                        ;;and  al, 3Fh                  ; enable IRQ 14 and IRQ 15
                                        ;;;JMP $+2
                                        ;;IODELAY
                                        ;;OUT  INTB01,AL
                                        ;;IODELAY
                                        ;;IN   AL,INTA01          ; LET INTERRUPTS PASS THRU TO
                                        ;;AND  AL,0FBH            ;  SECOND CHIP
                                        ;;;JMP $+2
                                        ;;IODELAY
                                        ;;OUT  INTA01,AL
                                        ;
                                        ;STI
                                        ;;PUSH DS                 ; MOVE ABS0 POINTER TO
                                        ;;POP  ES                 ; EXTRA SEGMENT POINTER
                                        ;;;CALL      DDS                ; ESTABLISH DATA SEGMENT
                                        ;;MOV  byte [DISK_STATUS1],0     ; RESET THE STATUS INDICATOR
                                        ;;MOV  byte [HF_NUM],0           ; ZERO NUMBER OF FIXED DISKS
                                        ;;MOV  byte [CONTROL_BYTE],0
                                        ;;MOV  byte [PORT_OFF],0   ; ZERO CARD OFFSET
                                        ; 20/12/2014 - private code by Erdogan Tan
                                                    ; (out of original PC-AT, PC-XT BIOS code)
                                        ;mov   si, hd0_type
00002CF4 BE[18490000]                   mov    esi, hd0_type
                                        ;mov   cx, 4
00002CF9 B904000000                     mov    ecx, 4
                                hde_l:
00002CFE AC                             lodsb
```

```
8644 00002CFF 3C80                      cmp    al, 80h                  ; 8?h = existing
8645 00002D01 7208                      jb     short _L4
8646 00002D03 FE05[B02C0000]            inc    byte [HF_NUM]       ; + 1 hard (fixed) disk drives
8647 00002D09 E2F3                      loop   hde_l
8648                                 _L4:                            ; 0 <= [HF_NUM] =< 4
8649                                 ;L4:
8650                                        ;
8651                                        ;; 31/12/2014 - cancel controller diagnostics here
8652                                        ;;;mov       cx, 3  ; 26/12/2014 (Award BIOS 1999)
8653                                        ;;mov  cl, 3
8654                                        ;;
8655                                        ;;MOV  DL,80H             ; CHECK THE CONTROLLER
8656                                 ;;hdc_dl:
8657                                        ;;MOV  AH,14H             ; USE CONTROLLER DIAGNOSTIC COMMAND
8658                                        ;;INT  13H                ; CALL BIOS WITH DIAGNOSTIC COMMAND
8659                                        ;;;JC  short CTL_ERRX            ; DISPLAY ERROR MESSAGE IF BAD RETURN
8660                                        ;;;jc  short POD_DONE ;22/12/2014
8661                                        ;;jnc  short hdc_reset0
8662                                        ;;loop hdc_dl
8663                                        ;;; 27/12/2014
8664                                        ;;stc
8665                                        ;;retn
8666                                        ;
8667                                 ;;hdc_reset0:
8668                                        ; 18/01/2015
8669 00002D0B 8A0D[B02C0000]               mov    cl, [HF_NUM]
8670 00002D11 20C9                         and    cl, cl
8671 00002D13 740D                         jz     short POD_DONE
8672                                       ;
8673 00002D15 B27F                         mov    dl, 7Fh
8674                                 hdc_reset1:
8675 00002D17 FEC2                         inc    dl
8676                                       ;; 31/12/2015
8677                                       ;;push dx
8678                                       ;;push cx
8679                                       ;;push ds
8680                                       ;;sub  ax, ax
8681                                       ;;mov  ds, ax
8682                                       ;;MOV  AX, [TIMER_LOW]           ; GET START TIMER COUNTS
8683                                       ;;pop  ds
8684                                       ;;MOV  BX,AX
8685                                       ;;ADD  AX,6*182           ; 60 SECONDS* 18.2
8686                                       ;;MOV  CX,AX
8687                                       ;;mov  word [wait_count], 0      ; 22/12/2014 (reset wait counter)
8688                                       ;;
8689                                       ;; 31/12/2014 - cancel HD_RESET_1
8690                                       ;;CALL HD_RESET_1         ; SET UP DRIVE 0, (1,2,3)
8691                                       ;;pop  cx
8692                                       ;;pop  dx
```

```
8693                                    ;;
8694                                    ; 18/01/2015
8695 00002D19 B40D                     mov   ah, 0Dh ; ALTERNATE RESET
8696                                    ;int  13h
8697 00002D1B E887FFFFFF               call  int13h
8698 00002D20 E2F5                     loop  hdc_reset1
8699                                POD_DONE:
8700 00002D22 C3                       RETn
8701
8702                                ;;-----     POD_ERROR
8703
8704                                ;;CTL_ERRX:
8705                                ;     ;MOV   SI,OFFSET F1782     ; CONTROLLER ERROR
8706                                ;     ;CALL  SET_FAIL            ; DO NOT IPL FROM DISK
8707                                ;     ;CALL  E_MSG               ; DISPLAY ERROR AND SET (BP) ERROR FLAG
8708                                ;     ;JMP   short POD_DONE
8709
8710                                ;;HD_RESET_1:
8711                                ;;    ;PUSH  BX                  ; SAVE TIMER LIMITS
8712                                ;;    ;PUSH  CX
8713                                ;;RES_1: MOV AH,09H              ; SET DRIVE PARAMETERS
8714                                ;;    INT    13H
8715                                ;;    JC     short RES_2
8716                                ;;    MOV    AH,11H              ; RECALIBRATE DRIVE
8717                                ;;    INT    13H
8718                                ;;    JNC    short RES_CK        ; DRIVE OK
8719                                ;;RES_2: ;CALL     POD_TCHK            ; CHECK TIME OUT
8720                                ;;    cmp    word [wait_count], 6*182 ; waiting time (in timer ticks)
8721                                ;;                                ; (30 seconds)
8722                                ;;    ;cmc
8723                                ;;    ;JNC   short RES_1
8724                                ;;    jb     short RES_1
8725                                ;;;RES_FL: ;MOV    SI,OFFSET F1781     ; INDICATE DISK 1 FAILURE;
8726                                ;;    ;TEST  DL,1
8727                                ;;    ;JNZ   RES_E1
8728                                ;;    ;MOV   SI,OFFSET F1780     ; INDICATE DISK 0 FAILURE
8729                                ;;    ;CALL  SET_FAIL            ; DO NOT TRY TO IPL DISK 0
8730                                ;;    ;JMP   SHORT RES_E1
8731                                ;;RES_ER: ; 22/12/2014
8732                                ;;RES_OK:
8733                                ;;    ;POP   CX                  ; RESTORE TIMER LIMITS
8734                                ;;    ;POP   BX
8735                                ;;    RETn
8736                                ;;
8737                                ;;RES_RS: MOV      AH,00H              ; RESET THE DRIVE
8738                                ;;    INT    13H
8739                                ;;RES_CK: MOV      AH,08H              ; GET MAX CYLINDER,HEAD,SECTOR
8740                                ;;    MOV    BL,DL               ; SAVE DRIVE CODE
8741                                ;;    INT    13H
```

```
8742                    ;;    JC     short RES_ER
8743                    ;;    MOV    word [NEC_STATUS],CX      ; SAVE MAX CYLINDER, SECTOR
8744                    ;;    MOV    DL,BL               ; RESTORE DRIVE CODE
8745                    ;;RES_3: MOV AX,0401H            ; VERIFY THE LAST SECTOR
8746                    ;;    INT    13H
8747                    ;;    JNC    short RES_OK        ; VERIFY OK
8748                    ;;    CMP    AH,BAD_SECTOR       ; OK ALSO IF JUST ID READ
8749                    ;;    JE     short RES_OK
8750                    ;;    CMP    AH,DATA_CORRECTED
8751                    ;;    JE     short RES_OK
8752                    ;;    CMP    AH,BAD_ECC
8753                    ;;    JE     short RES_OK
8754                    ;;    ;CALL  POD_TCHK            ; CHECK FOR TIME OUT
8755                    ;;    cmp    word [wait_count], 6*182 ; waiting time (in timer ticks)
8756                    ;;                               ; (60 seconds)
8757                    ;;    cmc
8758                    ;;    JC     short RES_ER        ; FAILED
8759                    ;;    MOV    CX,word [NEC_STATUS]      ; GET SECTOR ADDRESS, AND CYLINDER
8760                    ;;    MOV    AL,CL               ; SEPARATE OUT SECTOR NUMBER
8761                    ;;    AND    AL,3FH
8762                    ;;    DEC    AL                  ; TRY PREVIOUS ONE
8763                    ;;    JZ     short RES_RS        ; WE'VE TRIED ALL SECTORS ON TRACK
8764                    ;;    AND    CL,0C0H             ; KEEP CYLINDER BITS
8765                    ;;    OR     CL,AL               ; MERGE SECTOR WITH CYLINDER BITS
8766                    ;;    MOV    word [NEC_STATUS],CX      ; SAVE CYLINDER, NEW SECTOR NUMBER
8767                    ;;    JMP    short RES_3         ; TRY AGAIN
8768                    ;;;RES_ER: MOV     SI,OFFSET F1791     ; INDICATE DISK 1 ERROR
8769                    ;;    ;TEST  DL,1
8770                    ;;    ;JNZ   short RES_E1
8771                    ;;    ;MOV   SI,OFFSET F1790     ; INDICATE DISK 0 ERROR
8772                    ;;;RES_E1:
8773                    ;;    ;CALL  E_MSG               ; DISPLAY ERROR AND SET (BP) ERROR FLAG
8774                    ;;;RES_OK:
8775                    ;;    ;POP   CX                  ; RESTORE TIMER LIMITS
8776                    ;;    ;POP   BX
8777                    ;;    ;RETn
8778                    ;
8779                    ;;SET_FAIL:
8780                    ;     ;MOV   AX,X*(CMOS_DIAG+NMI)      ; GET CMOS ERROR BYTE
8781                    ;     ;CALL  CMOS_READ
8782                    ;     ;OR    AL,HF_FAIL          ; SET DO NOT IPL FROM DISK FLAG
8783                    ;     ;XCHG  AH,AL               ; SAVE IT
8784                    ;     ;CALL  CMOS_WRITE          ; PUT IT OUT
8785                    ;     ;RETn
8786                    ;
8787                    ;;POD_TCHK:                      ; CHECK FOR 30 SECOND TIME OUT
8788                    ;     ;POP   AX                  ; SAVE RETURN
8789                    ;     ;POP   CX                  ; GET TIME OUT LIMITS
8790                    ;     ;POP   BX
```

```
8791                                    ;     ;PUSH  BX                 ; AND SAVE THEM AGAIN
8792                                    ;     ;PUSH  CX
8793                                    ;     ;PUSH  AX
8794                                    ;     ;push  ds
8795                                    ;     ;xor   ax, ax
8796                                    ;     ;mov   ds, ax             ; RESTORE RETURN
8797                                    ;     ;MOV   AX, [TIMER_LOW]          ; AX = CURRENT TIME
8798                                    ;     ;                         ; BX = START TIME
8799                                    ;     ;                         ; CX = END TIME
8800                                    ;     ;pop   ds
8801                                    ;     ;CMP   BX,CX
8802                                    ;     ;JB    short TCHK1        ; START < END
8803                                    ;     ;CMP   BX,AX
8804                                    ;     ;JB    short TCHKG        ; END < START < CURRENT
8805                                    ;     ;JMP   SHORT TCHK2        ; END, CURRENT < START
8806                                    ;;TCHK1: CMP AX,BX
8807                                    ;;    JB     short TCHKNG       ; CURRENT < START < END
8808                                    ;;TCHK2: CMP AX,CX
8809                                    ;;    JB     short TCHKG        ; START < CURRENT < END
8810                                    ;;                              ; OR CURRENT < END < START
8811                                    ;;TCHKNG: STC                         ; CARRY SET INDICATES TIME OUT
8812                                    ;;    RETn
8813                                    ;;TCHKG: CLC                    ; INDICATE STILL TIME
8814                                    ;;    RETn
8815                                    ;;
8816                                    ;;int_13h:
8817
8818                                    ;----------------------------------------
8819                                    ;     FIXED DISK BIOS ENTRY POINT      :
8820                                    ;----------------------------------------
8821
8822                                    DISK_IO:
8823 00002D23 80FA80                         CMP    DL,80H             ; TEST FOR FIXED DISK DRIVE
8824                                          ;JAE   short A1           ; YES, HANDLE HERE
8825                                          ;;;INT 40H                ; DISKETTE HANDLER
8826                                          ;;call int40h
8827 00002D26 0F8223F1FFFF                    jb     DISKETTE_IO_1
8828                                    ;RET_2:
8829                                          ;RETf  2                  ; BACK TO CALLER
8830                                    ;     retf   4
8831                                    A1:
8832 00002D2C FB                              STI                       ; ENABLE INTERRUPTS
8833                                          ;; 04/01/2015
8834                                          ;;OR   AH,AH
8835                                          ;;JNZ  short A2
8836                                          ;;INT  40H                ; RESET NEC WHEN AH=0
8837                                          ;;SUB  AH,AH
8838 00002D2D 80FA83                         CMP    DL,(80H + S_MAX_FILE - 1)
8839 00002D30 772C                           JA     short RET_2
```

```
8840                                    ; 18/01/2015
8841 00002D32 08E4                      or     ah,ah
8842 00002D34 742B                      jz     short A4
8843 00002D36 80FC0D                    cmp    ah, 0Dh      ; Alternate reset
8844 00002D39 7504                      jne    short A2
8845 00002D3B 28E4                      sub    ah,ah  ; Reset
8846 00002D3D EB22                      jmp    short A4
8847                                A2:
8848 00002D3F 80FC08                    CMP    AH,08H              ; GET PARAMETERS IS A SPECIAL CASE
8849                                    ;JNZ   short A3
8850                                      ;JMP    GET_PARM_N
8851 00002D42 0F841C030000              je     GET_PARM_N
8852 00002D48 80FC15                A3: CMP    AH,15H              ; READ DASD TYPE IS ALSO
8853                                    ;JNZ   short A4
8854                                      ;JMP    READ_DASD_TYPE
8855 00002D4B 0F84C7020000                je     READ_DASD_TYPE
8856                                    ; 02/02/2015
8857 00002D51 80FC1D                    cmp    ah, 1Dh                  ;(Temporary for Retro UNIX 386 v1)
8858                                    ; 12/01/2015
8859 00002D54 F5                        cmc
8860 00002D55 730A                      jnc    short A4
8861                                    ; 30/01/2015
8862                                    ;mov     byte [CS:DISK_STATUS1],BAD_CMD  ; COMMAND ERROR
8863 00002D57 C605[AF2C0000]01            mov     byte [DISK_STATUS1], BAD_CMD
8864                                    ;jmp   short RET_2
8865                                RET_2:
8866 00002D5E CA0400                    retf   4
8867                                A4:                             ; SAVE REGISTERS DURING OPERATION
8868 00002D61 C8080000                  ENTER  8,0                  ; SAVE (BP) AND MAKE ROOM FOR @CMD_BLOCK
8869 00002D65 53                        PUSH   eBX                  ;  IN THE STACK, THE COMMAND BLOCK IS:
8870 00002D66 51                        PUSH   eCX                  ;   @CMD_BLOCK == BYTE PTR [BP]-8
8871 00002D67 52                        PUSH   eDX
8872 00002D68 1E                        PUSH   DS
8873 00002D69 06                        PUSH   ES
8874 00002D6A 56                        PUSH   eSI
8875 00002D6B 57                        PUSH   eDI
8876                                    ;;04/01/2015
8877                                    ;;OR   AH,AH                ; CHECK FOR RESET
8878                                    ;;JNZ  short A5
8879                                    ;;MOV  DL,80H               ; FORCE DRIVE 80 FOR RESET
8880                                ;;A5:
8881                                    ;push  cs
8882                                    ;pop   ds
8883                                    ; 21/02/2015
8884 00002D6C 6650                      push   ax
8885 00002D6E 66B81000                  mov    ax, KDATA
8886 00002D72 8ED8                      mov    ds, ax
8887 00002D74 8EC0                      mov    es, ax
8888 00002D76 6658                      pop    ax
```

```
8889 00002D78 E889000000                CALL   DISK_IO_CONT        ; PERFORM THE OPERATION
8890                                    ;;CALL DDS                  ; ESTABLISH SEGMENT
8891 00002D7D 8A25[AF2C0000]            MOV    AH,[DISK_STATUS1]    ; GET STATUS FROM OPERATION
8892 00002D83 80FC01                    CMP    AH,1                 ; SET THE CARRY FLAG TO INDICATE
8893 00002D86 F5                        CMC                         ; SUCCESS OR FAILURE
8894 00002D87 5F                        POP    eDI                  ; RESTORE REGISTERS
8895 00002D88 5E                        POP    eSI
8896 00002D89 07                          POP     ES
8897 00002D8A 1F                          POP     DS
8898 00002D8B 5A                        POP    eDX
8899 00002D8C 59                        POP    eCX
8900 00002D8D 5B                        POP    eBX
8901 00002D8E C9                        LEAVE                       ; ADJUST (SP) AND RESTORE (BP)
8902                                    ;RETf  2                    ; THROW AWAY SAVED FLAGS
8903 00002D8F CA0400                    retf   4
8904                                ; 21/02/2015
8905                                ;       dw --> dd
8906                                M1:                             ; FUNCTION TRANSFER TABLE
8907 00002D92 [542F0000]                dd     DISK_RESET           ; 000H
8908 00002D96 [CB2F0000]                dd     RETURN_STATUS        ; 001H
8909 00002D9A [D82F0000]                dd     DISK_READ            ; 002H
8910 00002D9E [E12F0000]                dd     DISK_WRITE           ; 003H
8911 00002DA2 [EA2F0000]                dd     DISK_VERF            ; 004H
8912 00002DA6 [02300000]                dd     FMT_TRK              ; 005H
8913 00002DAA [4A2F0000]                dd     BAD_COMMAND          ; 006H FORMAT BAD SECTORS
8914 00002DAE [4A2F0000]                dd     BAD_COMMAND          ; 007H FORMAT DRIVE
8915 00002DB2 [4A2F0000]                dd     BAD_COMMAND          ; 008H RETURN PARAMETERS
8916 00002DB6 [C9300000]                dd     INIT_DRV             ; 009H
8917 00002DBA [28310000]                dd     RD_LONG              ; 00AH
8918 00002DBE [31310000]                dd     WR_LONG              ; 00BH
8919 00002DC2 [3A310000]                dd     DISK_SEEK            ; 00CH
8920 00002DC6 [542F0000]                dd     DISK_RESET           ; 00DH
8921 00002DCA [4A2F0000]                dd     BAD_COMMAND          ; 00EH READ BUFFER
8922 00002DCE [4A2F0000]                dd     BAD_COMMAND          ; 00FH WRITE BUFFER
8923 00002DD2 [62310000]                dd     TST_RDY              ; 010H
8924 00002DD6 [86310000]                dd     HDISK_RECAL          ; 011H
8925 00002DDA [4A2F0000]                dd     BAD_COMMAND          ; 012H MEMORY DIAGNOSTIC
8926 00002DDE [4A2F0000]                dd     BAD_COMMAND          ; 013H DRIVE DIAGNOSTIC
8927 00002DE2 [BC310000]                dd     CTLR_DIAGNOSTIC      ; 014H CONTROLLER DIAGNOSTIC
8928                                    ; 02/02/2015 (Temporary - Retro UNIX 386 v1 - DISK I/O test)
8929 00002DE6 [4A2F0000]                dd     BAD_COMMAND          ; 015h
8930 00002DEA [4A2F0000]                dd     BAD_COMMAND          ; 016h
8931 00002DEE [4A2F0000]                dd     BAD_COMMAND          ; 017h
8932 00002DF2 [4A2F0000]                dd     BAD_COMMAND          ; 018h
8933 00002DF6 [4A2F0000]                dd     BAD_COMMAND          ; 019h
8934 00002DFA [4A2F0000]                dd     BAD_COMMAND          ; 01Ah
8935 00002DFE [D82F0000]                dd     DISK_READ            ; 01Bh ; LBA read
8936 00002E02 [E12F0000]                dd     DISK_WRITE           ; 01Ch ; LBA write
8937                                M1L     EQU    $-M1
```

```
8938
8939                                    DISK_IO_CONT:
8940                                          ;;CALL DDS                ; ESTABLISH SEGMENT
8941 00002E06 80FC01                         CMP    AH,01H             ; RETURN STATUS
8942                                          ;;JNZ  short SU0
8943                                            ;;JMP    RETURN_STATUS
8944 00002E09 0F84BC010000                   je     RETURN_STATUS
8945                                    SU0:
8946 00002E0F C605[AF2C0000]00               MOV    byte [DISK_STATUS1],0     ; RESET THE STATUS INDICATOR
8947                                          ;;PUSH BX                 ; SAVE DATA ADDRESS
8948                                          ;mov   si, bx ;; 14/02/2015
8949 00002E16 89DE                           mov    esi, ebx ; 21/02/2015
8950 00002E18 8A1D[B02C0000]                 MOV    BL,[HF_NUM]        ; GET NUMBER OF DRIVES
8951                                          ;; 04/01/2015
8952                                          ;;PUSH AX
8953 00002E1E 80E27F                         AND    DL,7FH             ; GET DRIVE AS 0 OR 1
8954                                                                   ; (get drive number as 0 to 3)
8955 00002E21 38D3                           CMP    BL,DL
8956                                            ;;JBE   BAD_COMMAND_POP         ; INVALID DRIVE
8957 00002E23 0F8621010000                     jbe     BAD_COMMAND ;; 14/02/2015
8958                                            ;
8959                                          ;;03/01/2015
8960 00002E29 29DB                           sub    ebx, ebx
8961 00002E2B 88D3                           mov    bl, dl
8962                                          ;sub   bh, bh
8963 00002E2D 883D[31480000]                 mov    [LBAMode], bh      ; 0
8964                                          ;;test byte [bx+hd0_type], 1     ; LBA ready ?
8965                                          ;test  byte [ebx+hd0_type], 1
8966                                          ;jz    short su1          ; no
8967                                          ;inc   byte [LBAMode]
8968                                    ;su1:
8969                                          ; 21/02/2015 (32 bit modification)
8970                                          ;04/01/2015
8971 00002E33 6650                           push   ax ; ***
8972                                          ;PUSH  ES ; **
8973 00002E35 6652                           PUSH   DX ; *
8974 00002E37 6650                           push   ax
8975 00002E39 E865060000                     CALL   GET_VEC            ; GET DISK PARAMETERS
8976                                          ; 02/02/2015
8977                                          ;mov   ax, [ES:BX+16] ; I/O port base address (1F0h, 170h)
8978 00002E3E 668B4310                       mov    ax, [ebx+16]
8979 00002E42 66A3[B62C0000]                 mov    [HF_PORT], ax
8980                                          ;mov   dx, [ES:BX+18] ; control port address (3F6h, 376h)
8981 00002E48 668B5312                       mov    dx, [ebx+18]
8982 00002E4C 668915[B82C0000]               mov    [HF_REG_PORT], dx
8983                                          ;mov   al, [ES:BX+20] ; head register upper nibble (A0h,B0h,E0h,F0h)
8984 00002E53 8A4314                         mov    al, [ebx+20]
8985                                          ; 23/02/2015
8986 00002E56 A840                           test   al, 40h       ; LBA bit (bit 6)
```

```
8987 00002E58 7406                      jz      short su1
8988 00002E5A FE05[31480000]            inc     byte [LBAMode] ; 1
8989                                 su1:
8990 00002E60 C0E804                    shr     al, 4
8991 00002E63 2401                      and     al, 1
8992 00002E65 A2[BA2C0000]              mov     [hf_m_s], al
8993                                    ;
8994                                    ; 03/01/2015
8995                                    ;MOV   AL,byte [ES:BX+8]   ; GET CONTROL BYTE MODIFIER
8996 00002E6A 8A4308                    mov     al, [ebx+8]
8997                                    ;MOV   DX,[HF_REG_PORT]    ; Device Control register
8998 00002E6D EE                        OUT     DX,AL              ; SET EXTRA HEAD OPTION
8999                                                               ; Control Byte:  (= 08h, here)
9000                                                               ; bit 0 - 0
9001                                                               ; bit 1 - nIEN (1 = disable irq)
9002                                                               ; bit 2 - SRST (software RESET)
9003                                                               ; bit 3 - use extra heads (8 to 15)
9004                                                               ;         -always set to 1-
9005                                                               ; (bits 3 to 7 are reserved
9006                                                               ;          for ATA devices)
9007 00002E6E 8A25[B12C0000]            MOV     AH,[CONTROL_BYTE]  ; SET EXTRA HEAD OPTION IN
9008 00002E74 80E4C0                    AND     AH,0C0H            ; CONTROL BYTE
9009 00002E77 08C4                      OR      AH,AL
9010 00002E79 8825[B12C0000]            MOV     [CONTROL_BYTE],AH
9011                                    ; 04/01/2015
9012 00002E7F 6658                      pop     ax
9013 00002E81 665A                      pop     dx ; * ;; 14/02/2015
9014 00002E83 20E4                      and     ah, ah ; Reset function ?
9015 00002E85 7507                      jnz     short su2
9016                                    ;;pop  dx ; * ;; 14/02/2015
9017                                    ;pop   es ; **
9018 00002E87 6658                      pop     ax ; ***
9019                                    ;;pop  bx
9020 00002E89 E9C6000000                  jmp     DISK_RESET
9021                                 su2:
9022 00002E8E 803D[31480000]00          cmp     byte [LBAMode], 0
9023 00002E95 7661                      jna     short su3
9024                                    ;
9025                                    ; 02/02/2015 (LBA read/write function calls)
9026 00002E97 80FC1B                    cmp     ah, 1Bh
9027 00002E9A 720B                      jb      short lbarw1
9028 00002E9C 80FC1C                    cmp     ah, 1Ch
9029 00002E9F 775C                      ja      short invldfnc
9030                                    ;;pop  dx ; * ; 14/02/2015
9031                                    ;mov   ax, cx ; Lower word of LBA address (bits 0-15)
9032 00002EA1 89C8                      mov     eax, ecx ; LBA address (21/02/2015)
9033                                    ;; 14/02/2015
9034 00002EA3 88D1                      mov     cl, dl ; 14/02/2015
9035                                    ;;mov  dx, bx
```

```
9036                                    ;mov   dx, si ; higher word of LBA address (bits 16-23)
9037                                    ;;mov  bx, di
9038                                    ;mov   si, di ; Buffer offset
9039 00002EA5 EB31                      jmp    short lbarw2
9040                                lbarw1:
9041                                    ; convert CHS to LBA
9042                                    ;
9043                                    ; LBA calculation - AWARD BIOS - 1999 - AHDSK.ASM
9044                                    ; LBA = "# of Heads" * Sectors/Track * Cylinder + Head * Sectors/Track
9045                                    ;      + Sector - 1
9046 00002EA7 6652                      push   dx ; * ;; 14/02/2015
9047                                    ;xor   dh, dh
9048 00002EA9 31D2                      xor    edx, edx
9049                                    ;mov   dl, [ES:BX+14]      ; sectors per track (logical)
9050 00002EAB 8A530E                    mov    dl, [ebx+14]
9051                                    ;xor   ah, ah
9052 00002EAE 31C0                      xor    eax, eax
9053                                    ;mov   al, [ES:BX+2]; heads (logical)
9054 00002EB0 8A4302                    mov    al, [ebx+2]
9055 00002EB3 FEC8                      dec    al
9056 00002EB5 6640                      inc    ax           ; 0 =  256
9057 00002EB7 66F7E2                    mul    dx
9058                                    ; AX = # of Heads" * Sectors/Track
9059 00002EBA 6689CA                    mov    dx, cx
9060                                    ;and   cx, 3Fh       ; sector  (1 to 63)
9061 00002EBD 83E13F                    and    ecx, 3fh
9062 00002EC0 86D6                      xchg   dl, dh
9063 00002EC2 C0EE06                    shr    dh, 6
9064                                    ; DX = cylinder (0 to 1023)
9065                                    ;mul   dx
9066                                    ; DX:AX = # of Heads" * Sectors/Track * Cylinder
9067 00002EC5 F7E2                      mul    edx
9068 00002EC7 FEC9                      dec    cl  ; sector - 1
9069                                    ;add   ax, cx
9070                                    ;adc   dx, 0
9071                                    ; DX:AX = # of Heads" * Sectors/Track * Cylinder + Sector -1
9072 00002EC9 01C8                      add    eax, ecx
9073 00002ECB 6659                      pop    cx ; * ; ch = head, cl = drive number (zero based)
9074                                    ;push  dx
9075                                    ;push  ax
9076 00002ECD 50                        push   eax
9077                                    ;mov   al, [ES:BX+14]      ; sectors per track (logical)
9078 00002ECE 8A430E                    mov    al, [ebx+14]
9079 00002ED1 F6E5                      mul    ch
9080                                    ;  AX = Head * Sectors/Track
9081 00002ED3 6699                        cwd
9082                                    ;pop   dx
9083 00002ED5 5A                        pop    edx
9084                                    ;add   ax, dx
```

```
9085                                    ;pop   dx
9086                                    ;adc   dx, 0 ; add carry bit
9087 00002ED6 01D0                      add    eax, edx
9088                                lbarw2:
9089 00002ED8 29D2                      sub    edx, edx ; 21/02/2015
9090 00002EDA 88CA                      mov    dl, cl ; 21/02/2015
9091 00002EDC C645F800                    mov     byte [CMD_BLOCK], 0 ; Features Register
9092                                                    ; NOTE: Features register (1F1h, 171h)
9093                                                    ; is not used for ATA device R/W functions.
9094                                                    ; It is old/obsolete 'write precompensation'
9095                                                    ; register and error register
9096                                                    ; for old ATA/IDE devices.
9097                                    ; 18/01/2014
9098                                    ;mov   ch, [hf_m_s] ; Drive 0 (master) or 1 (slave)
9099 00002EE0 8A0D[BA2C0000]            mov    cl, [hf_m_s]
9100                                    ;shl   ch, 4        ; bit 4 (drive bit)
9101                                    ;or    ch, 0E0h     ; bit 5 = 1
9102                                                        ; bit 6 = 1 = LBA mode
9103                                                        ; bit 7 = 1
9104 00002EE6 80C90E                    or     cl, 0Eh ; 1110b
9105                                    ;and   dh, 0Fh             ; LBA byte 4 (bits 24 to 27)
9106 00002EE9 25FFFFFF0F                and    eax, 0FFFFFFFh
9107 00002EEE C1E11C                    shl    ecx, 28 ; 21/02/2015
9108                                    ;or    dh, ch
9109 00002EF1 09C8                      or     eax, ecx
9110                                    ;;mov  [CMD_BLOCK+2], al ; LBA byte 1 (bits 0 to 7)
9111                                                             ; (Sector Number Register)
9112                                    ;;mov  [CMD_BLOCK+3], ah ; LBA byte 2 (bits 8 to 15)
9113                                                             ; (Cylinder Low Register)
9114                                    ;mov   [CMD_BLOCK+2], ax ; LBA byte 1, 2
9115                                    ;mov   [CMD_BLOCK+4], dl ; LBA byte 3 (bits 16 to 23)
9116                                                             ; (Cylinder High Register)
9117                                    ;;mov  [CMD_BLOCK+5], dh ; LBA byte 4 (bits 24 to 27)
9118                                                             ; (Drive/Head Register)
9119
9120                                    ;mov   [CMD_BLOCK+4], dx ; LBA byte 4, LBA & DEV select bits
9121 00002EF3 8945FA                    mov    [CMD_BLOCK+2], eax ; 21/02/2015
9122                                    ;14/02/2015
9123                                    ;mov   dl, cl ; Drive number (INIT_DRV)
9124 00002EF6 EB38                      jmp    short su4
9125                                su3:
9126                                    ; 02/02/2015
9127                                    ; (Temporary functions 1Bh & 1Ch are not valid for CHS mode)
9128 00002EF8 80FC14                    cmp    ah, 14h
9129 00002EFB 7604                      jna    short chsfnc
9130                                invldfnc:
9131                                      ; 14/02/2015
9132                                    ;pop   es ; **
9133 00002EFD 6658                        pop     ax ; ***
```

```
9134                                            ;jmp     short BAD_COMMAND_POP
9135 00002EFF EB49                              jmp     short BAD_COMMAND
9136                                    chsfnc:
9137                                          ;MOV   AX,[ES:BX+5]       ; GET WRITE PRE-COMPENSATION CYLINDER
9138 00002F01 668B4305                        mov    ax, [ebx+5]
9139 00002F05 66C1E802                        SHR    AX,2
9140 00002F09 8845F8                          MOV    [CMD_BLOCK],AL
9141                                          ;;MOV  AL,[ES:BX+8]       ; GET CONTROL BYTE MODIFIER
9142                                          ;;PUSH DX
9143                                          ;;MOV  DX,[HF_REG_PORT]
9144                                          ;;OUT  DX,AL              ; SET EXTRA HEAD OPTION
9145                                          ;;POP  DX ; *
9146                                          ;;POP  ES ; **
9147                                          ;;MOV  AH,[CONTROL_BYTE]  ; SET EXTRA HEAD OPTION IN
9148                                          ;;AND  AH,0C0H            ; CONTROL BYTE
9149                                          ;;OR   AH,AL
9150                                          ;;MOV  [CONTROL_BYTE],AH
9151                                          ;
9152 00002F0C 88C8                            MOV    AL,CL              ; GET SECTOR NUMBER
9153 00002F0E 243F                            AND    AL,3FH
9154 00002F10 8845FA                          MOV    [CMD_BLOCK+2],AL
9155 00002F13 886DFB                          MOV    [CMD_BLOCK+3],CH   ; GET CYLINDER NUMBER
9156 00002F16 88C8                            MOV    AL,CL
9157 00002F18 C0E806                          SHR    AL,6
9158 00002F1B 8845FC                          MOV    [CMD_BLOCK+4],AL   ; CYLINDER HIGH ORDER 2 BITS
9159                                          ;;05/01/2015
9160                                          ;;MOV  AL,DL              ; DRIVE NUMBER
9161 00002F1E A0[BA2C0000]                    mov    al, [hf_m_s]
9162 00002F23 C0E004                          SHL    AL,4
9163 00002F26 80E60F                          AND    DH,0FH             ; HEAD NUMBER
9164 00002F29 08F0                            OR     AL,DH
9165                                          ;OR    AL,80H or 20H
9166 00002F2B 0CA0                            OR     AL,80h+20h         ; ECC AND 512 BYTE SECTORS
9167 00002F2D 8845FD                          MOV    [CMD_BLOCK+5],AL   ; ECC/SIZE/DRIVE/HEAD
9168                                    su4:
9169                                          ;POP   ES ; **
9170                                            ;; 14/02/2015
9171                                            ;;POP   AX
9172                                            ;;MOV   [CMD_BLOCK+1],AL        ; SECTOR COUNT
9173                                            ;;PUSH  AX
9174                                            ;;MOV   AL,AH                   ; GET INTO LOW BYTE
9175                                            ;;XOR   AH,AH                   ; ZERO HIGH BYTE
9176                                            ;;SAL   AX,1                    ; *2 FOR TABLE LOOKUP
9177 00002F30 6658                              pop     ax ; ***
9178 00002F32 8845F9                            mov     [CMD_BLOCK+1], al
9179 00002F35 29DB                              sub  ebx, ebx
9180 00002F37 88E3                            mov     bl, ah
9181                                            ;xor     bh, bh
9182                                            ;sal     bx, 1
```

```
9183 00002F39 66C1E302                    sal  bx, 2  ; 32 bit offset (21/02/2015)
9184                                      ;;MOV   SI,AX                   ; PUT INTO SI FOR BRANCH
9185                                        ;;CMP   AX,M1L                  ; TEST WITHIN RANGE
9186                                        ;;JNB   short BAD_COMMAND_POP
9187                                        ;cmp     bx, M1L
9188 00002F3D 83FB74                     cmp    ebx, M1L
9189 00002F40 7308                       jnb    short BAD_COMMAND
9190                                        ;xchg    bx, si
9191 00002F42 87DE                          xchg ebx, esi
9192                                      ;;;POP AX                 ; RESTORE AX
9193                                      ;;;POP BX                 ; AND DATA ADDRESS
9194
9195                                      ;;PUSH CX
9196                                      ;;PUSH AX                 ; ADJUST ES:BX
9197                                      ;MOV   CX,BX              ; GET 3 HIGH ORDER NIBBLES OF BX
9198                                      ;SHR   CX,4
9199                                      ;MOV   AX,ES
9200                                      ;ADD   AX,CX
9201                                      ;MOV   ES,AX
9202                                      ;AND   BX,000FH           ; ES:BX CHANGED TO ES:000X
9203                                      ;;POP  AX
9204                                      ;;POP  CX
9205                                      ;;JMP  word [CS:SI+M1]
9206                                      ;jmp   word [SI+M1]
9207 00002F44 FFA6[922D0000]             jmp    dword [esi+M1]
9208                                  ;;BAD_COMMAND_POP:
9209                                  ;;    POP    AX
9210                                  ;;    POP    BX
9211                                  BAD_COMMAND:
9212 00002F4A C605[AF2C0000]01                MOV     byte [DISK_STATUS1],BAD_CMD  ; COMMAND ERROR
9213 00002F51 B000                       MOV    AL,0
9214 00002F53 C3                         RETn
9215
9216                                  ;---------------------------------------
9217                                  ;     RESET THE DISK SYSTEM  (AH=00H) :
9218                                  ;---------------------------------------
9219
9220                                  ; 18-1-2015 : one controller reset (not other one)
9221
9222                                  DISK_RESET:
9223 00002F54 FA                         CLI
9224 00002F55 E4A1                       IN     AL,INTB01          ; GET THE MASK REGISTER
9225                                     ;JMP   $+2
9226                                     IODELAY
9227 00002F57 EB00                <1>  jmp short $+2
9228 00002F59 EB00                <1>  jmp short $+2
9229                                     ;AND   AL,0BFH             ; ENABLE FIXED DISK INTERRUPT
9230 00002F5B 243F                       and    al,3Fh              ; 22/12/2014 (IRQ 14 & IRQ 15)
9231 00002F5D E6A1                       OUT    INTB01,AL
```

```
9232 00002F5F FB                        STI                     ; START INTERRUPTS
9233                                    ; 14/02/2015
9234 00002F60 6689D7                    mov   di, dx
9235                                    ; 04/01/2015
9236                                    ;xor  di,di
9237                                drst0:
9238 00002F63 B004                      MOV   AL,04H  ; bit 2 - SRST
9239                                    ;MOV  DX,HF_REG_PORT
9240 00002F65 668B15[B82C0000]          MOV   DX,[HF_REG_PORT]
9241 00002F6C EE                        OUT   DX,AL             ; RESET
9242                                ;       MOV   CX,10             ; DELAY COUNT
9243                                ;DRD:   DEC   CX
9244                                ;       JNZ   short DRD         ; WAIT 4.8 MICRO-SEC
9245                                        ;mov  cx,2              ; wait for 30 micro seconds
9246 00002F6D B902000000                  mov  ecx, 2 ; 21/02/2015
9247 00002F72 E8DEEAFFFF                call   WAITF                    ; (Award Bios 1999 - WAIT_REFRESH,
9248                                                                      ; 40 micro seconds)
9249 00002F77 A0[B12C0000]              mov   al,[CONTROL_BYTE]
9250 00002F7C 240F                      AND   AL,0FH            ; SET HEAD OPTION
9251 00002F7E EE                        OUT   DX,AL             ; TURN RESET OFF
9252 00002F7F E80E040000                CALL  NOT_BUSY
9253 00002F84 7515                      JNZ   short DRERR       ; TIME OUT ON RESET
9254 00002F86 668B15[B62C0000]          MOV   DX,[HF_PORT]
9255 00002F8D FEC2                      inc   dl  ; HF_PORT+1
9256                                    ; 02/01/2015 - Award BIOS 1999 - AHDSK.ASM
9257                                      ;mov     cl, 10
9258 00002F8F B90A000000                  mov     ecx, 10 ; 21/02/2015
9259                                drst1:
9260 00002F94 EC                        IN    AL,DX             ; GET RESET STATUS
9261 00002F95 3C01                      CMP   AL,1
9262                                    ; 04/01/2015
9263 00002F97 740A                      jz    short drst2
9264                                    ;JNZ  short DRERR       ; BAD RESET STATUS
9265                                          ; Drive/Head Register - bit 4
9266 00002F99 E2F9                      loop  drst1
9267                                DRERR:
9268 00002F9B C605[AF2C0000]05          MOV   byte [DISK_STATUS1],BAD_RESET ; CARD FAILED
9269 00002FA2 C3                        RETn
9270                                drst2:
9271                                    ; 14/02/2015
9272 00002FA3 6689FA                    mov   dx,di
9273                                ;drst3:
9274                                ;       ; 05/01/2015
9275                                ;       shl   di,1
9276                                ;       ; 04/01/2015
9277                                ;       mov   ax,[di+hd_cports]
9278                                ;       cmp   ax,[HF_REG_PORT]
9279                                ;       je    short drst4
9280                                ;       mov   [HF_REG_PORT], ax
```

```
                                ;     ; 03/01/2015
                                ;     mov   ax,[di+hd_ports]
                                ;       mov     [HF_PORT], ax
                                ;     ; 05/01/2014
                                ;     shr   di,1
                                ;     ; 04/01/2015
                                ;     jmp   short drst0  ; reset other controller
                                ;drst4:
                                ;     ; 05/01/2015
                                ;     shr   di,1
                                ;     mov   al,[di+hd_dregs]
                                ;     and   al,10h ; bit 4 only
                                ;     shr   al,4 ; bit 4  -> bit 0
                                ;     mov   [hf_m_s], al ; (0 = master, 1 = slave)
                                      ;
00002FA6 A0[BA2C0000]                 mov   al, [hf_m_s] ; 18/01/2015
00002FAB A801                         test  al,1
                                ;     jnz   short drst6
00002FAD 7516                           jnz     short drst4
00002FAF 8065FDEF                     AND     byte [CMD_BLOCK+5],0EFH ; SET TO DRIVE 0
                                ;drst5:
                                drst3:
00002FB3 E811010000                   CALL  INIT_DRV           ; SET MAX HEADS
                                      ;mov  dx,di
00002FB8 E8C9010000                   CALL  HDISK_RECAL        ; RECAL TO RESET SEEK SPEED
                                      ; 04/01/2014
                                ;     inc   di
                                ;     mov   dx,di
                                ;     cmp   dl,[HF_NUM]
                                ;     jb    short drst3
                                ;DRE:
00002FBD C605[AF2C0000]00             MOV   byte [DISK_STATUS1],0     ; IGNORE ANY SET UP ERRORS
00002FC4 C3                           RETn
                                ;drst6:
                                drst4:            ; Drive/Head Register - bit 4
00002FC5 804DFD10                     OR      byte [CMD_BLOCK+5],010H ; SET TO DRIVE 1
                                        ;jmp    short drst5
00002FC9 EBE8                           jmp     short drst3

                                ;--------------------------------------
                                ;     DISK STATUS ROUTINE  (AH = 01H) :
                                ;--------------------------------------

                                RETURN_STATUS:
00002FCB A0[AF2C0000]                 MOV   AL,[DISK_STATUS1]   ; OBTAIN PREVIOUS STATUS
00002FD0 C605[AF2C0000]00               MOV     byte [DISK_STATUS1],0   ; RESET STATUS
00002FD7 C3                           RETn

                                ;--------------------------------------
```

```
9330                                    ;     DISK READ ROUTINE    (AH = 02H) :
9331                                    ;----------------------------------------
9332
9333                                    DISK_READ:
9334 00002FD8 C645FE20                       MOV    byte [CMD_BLOCK+6],READ_CMD
9335 00002FDC E930020000                       JMP     COMMANDI
9336
9337                                    ;----------------------------------------
9338                                    ;     DISK WRITE ROUTINE   (AH = 03H) :
9339                                    ;----------------------------------------
9340
9341                                    DISK_WRITE:
9342 00002FE1 C645FE30                       MOV    byte [CMD_BLOCK+6],WRITE_CMD
9343 00002FE5 E97C020000                       JMP     COMMANDO
9344
9345                                    ;----------------------------------------
9346                                    ;     DISK VERIFY       (AH = 04H) :
9347                                    ;----------------------------------------
9348
9349                                    DISK_VERF:
9350 00002FEA C645FE40                       MOV    byte [CMD_BLOCK+6],VERIFY_CMD
9351 00002FEE E8EA020000                     CALL   COMMAND
9352 00002FF3 750C                           JNZ    short VERF_EXIT           ; CONTROLLER STILL BUSY
9353 00002FF5 E85C030000                     CALL   _WAIT               ; (Original: CALL WAIT)
9354 00002FFA 7505                           JNZ    short VERF_EXIT           ; TIME OUT
9355 00002FFC E8E9030000                     CALL   CHECK_STATUS
9356                                    VERF_EXIT:
9357 00003001 C3                             RETn
9358
9359                                    ;----------------------------------------
9360                                    ;     FORMATTING        (AH = 05H) :
9361                                    ;----------------------------------------
9362
9363                                    FMT_TRK:                          ; FORMAT TRACK      (AH = 005H)
9364 00003002 C645FE50                       MOV    byte [CMD_BLOCK+6],FMTTRK_CMD
9365                                          ;PUSH  ES
9366                                          ;PUSH  BX
9367 00003006 53                             push   ebx
9368 00003007 E897040000                     CALL   GET_VEC             ; GET DISK PARAMETERS ADDRESS
9369                                          ;MOV   AL,[ES:BX+14]       ; GET SECTORS/TRACK
9370 0000300C 8A430E                         mov    al, [ebx+14]
9371 0000300F 8845F9                         MOV    [CMD_BLOCK+1],AL    ; SET SECTOR COUNT IN COMMAND
9372 00003012 5B                             pop    ebx
9373                                          ;POP   BX
9374                                          ;POP   ES
9375 00003013 E955020000                       JMP     CMD_OF                  ; GO EXECUTE THE COMMAND
9376
9377                                    ;----------------------------------------
9378                                    ;     READ DASD TYPE           (AH = 15H) :
```

```
                                ;---------------------------------------

                                READ_DASD_TYPE:
                                READ_D_T:                       ; GET DRIVE PARAMETERS
00003018 1E                           PUSH   DS                 ; SAVE REGISTERS
                                      ;PUSH  ES
00003019 53                           PUSH   eBX
                                      ;CALL  DDS                ; ESTABLISH ADDRESSING
                                      ;push  cs
                                      ;pop   ds
0000301A 66BB1000                       mov  bx, KDATA
0000301E 8EDB                         mov    ds, bx
                                      ;mov   es, bx
00003020 C605[AF2C0000]00             MOV     byte [DISK_STATUS1],0
00003027 8A1D[B02C0000]               MOV    BL,[HF_NUM]        ; GET NUMBER OF DRIVES
0000302D 80E27F                       AND    DL,7FH             ; GET DRIVE NUMBER
00003030 38D3                         CMP    BL,DL
00003032 7625                         JBE    short RDT_NOT_PRESENT     ; RETURN DRIVE NOT PRESENT
00003034 E86A040000                   CALL   GET_VEC            ; GET DISK PARAMETER ADDRESS
                                      ;MOV   AL,[ES:BX+2]       ; HEADS
00003039 8A4302                       mov    al, [ebx+2]
                                      ;MOV   CL,[ES:BX+14]
0000303C 8A4B0E                       mov    cl, [ebx+14]
0000303F F6E9                         IMUL   CL                 ; * NUMBER OF SECTORS
                                      ;MOV   CX,[ES:BX]         ; MAX NUMBER OF CYLINDERS
00003041 668B0B                       mov    cx ,[ebx]
                                      ;
                                      ; 02/01/2015
                                      ; ** leave the last cylinder as reserved for diagnostics **
                                      ; (Also in Award BIOS - 1999, AHDSK.ASM, FUN15 -> sub ax, 1)
00003044 6649                         DEC    CX                 ; LEAVE ONE FOR DIAGNOSTICS
                                      ;
00003046 66F7E9                       IMUL   CX                 ; NUMBER OF SECTORS
00003049 6689D1                       MOV    CX,DX              ; HIGH ORDER HALF
0000304C 6689C2                       MOV    DX,AX              ; LOW ORDER HALF
                                      ;SUB   AX,AX
0000304F 28C0                         sub    al, al
00003051 B403                         MOV    AH,03H             ; INDICATE FIXED DISK
00003053 5B                     RDT2: POP    eBX                ; RESTORE REGISTERS
                                      ;POP   ES
00003054 1F                           POP    DS
00003055 F8                           CLC                       ; CLEAR CARRY
                                      ;RETf  2
00003056 CA0400                       retf   4
                                RDT_NOT_PRESENT:
00003059 6629C0                       SUB    AX,AX              ; DRIVE NOT PRESENT RETURN
0000305C 6689C1                       MOV    CX,AX              ; ZERO BLOCK COUNT
0000305F 6689C2                       MOV    DX,AX
00003062 EBEF                         JMP    short RDT2
```

```
9428
9429                                    ;----------------------------------------
9430                                    ;     GET PARAMETERS          (AH = 08H) :
9431                                    ;----------------------------------------
9432
9433                                    GET_PARM_N:
9434                                    ;GET_PARM:                      ; GET DRIVE PARAMETERS
9435 00003064 1E                               PUSH   DS                ; SAVE REGISTERS
9436                                           ;PUSH  ES
9437 00003065 53                               PUSH   eBX
9438                                           ;MOV   AX,ABS0           ; ESTABLISH ADDRESSING
9439                                           ;MOV   DS,AX
9440                                           ;TEST  DL,1              ; CHECK FOR DRIVE 1
9441                                           ;JZ    short G0
9442                                           ;LES   BX,@HF1_TBL_VEC
9443                                           ;JMP   SHORT G1
9444                                    ;G0:   LES    BX,@HF_TBL_VEC
9445                                    ;G1:
9446                                           ;CALL  DDS               ; ESTABLISH SEGMENT
9447                                           ; 22/12/2014
9448                                           ;push  cs
9449                                           ;pop   ds
9450 00003066 66BB1000                         mov    bx, KDATA
9451 0000306A 8EDB                             mov    ds, bx
9452                                           ;mov   es, bx
9453                                           ;
9454 0000306C 80EA80                           SUB    DL,80H
9455 0000306F 80FA04                           CMP    DL,MAX_FILE       ; TEST WITHIN RANGE
9456 00003072 7341                             JAE    short G4
9457                                           ;
9458 00003074 31DB                             xor    ebx, ebx ; 21/02/2015
9459                                           ; 22/12/2014
9460 00003076 88D3                             mov    bl, dl
9461                                           ;xor   bh, bh
9462 00003078 C0E302                           shl    bl, 2             ; convert index to offset
9463                                           ;add   bx, HF_TBL_VEC
9464 0000307B 81C3[BC2C0000]                   add    ebx, HF_TBL_VEC
9465                                           ;mov   ax, [bx+2]
9466                                           ;mov   es, ax            ; dpt segment
9467                                           ;mov   bx, [bx]          ; dpt offset
9468 00003081 8B1B                             mov    ebx, [ebx] ; 32 bit offset
9469
9470 00003083 C605[AF2C0000]00                 MOV    byte [DISK_STATUS1],0
9471                                             ;MOV     AX,[ES:BX]             ; MAX NUMBER OF CYLINDERS
9472 0000308A 668B03                           mov    ax, [ebx]
9473                                           ;;SUB  AX,2              ; ADJUST FOR 0-N
9474 0000308D 6648                             dec    ax                ; max. cylinder number
9475 0000308F 88C5                             MOV    CH,AL
9476 00003091 66250003                         AND    AX,0300H          ; HIGH TWO BITS OF CYLINDER
```

```
9477 00003095 66D1E8                    SHR    AX,1
9478 00003098 66D1E8                    SHR    AX,1
9479                                    ;OR    AL,[ES:BX+14]       ; SECTORS
9480 0000309B 0A430E                    or     al, [ebx+14]
9481 0000309E 88C1                      MOV    CL,AL
9482                                    ;MOV   DH,[ES:BX+2]        ; HEADS
9483 000030A0 8A7302                    mov    dh, [ebx+2]
9484 000030A3 FECE                      DEC    DH                  ; 0-N RANGE
9485 000030A5 8A15[B02C0000]            MOV    DL,[HF_NUM]         ; DRIVE COUNT
9486 000030AB 6629C0                    SUB    AX,AX
9487                                      ;27/12/2014
9488                                    ; ES:DI = Address of disk parameter table from BIOS
9489                                    ;(Programmer's Guide to the AMIBIOS - 1993)
9490                                    ;mov   di, bx              ; HDPT offset
9491 000030AE 89DF                      mov    edi, ebx
9492                                G5:
9493 000030B0 5B                        POP    eBX                 ; RESTORE REGISTERS
9494                                    ;POP   ES
9495 000030B1 1F                        POP    DS
9496                                    ;RETf  2
9497 000030B2 CA0400                    retf   4
9498                                G4:
9499 000030B5 C605[AF2C0000]07          MOV     byte [DISK_STATUS1],INIT_FAIL ; OPERATION FAILED
9500 000030BC B407                      MOV    AH,INIT_FAIL
9501 000030BE 28C0                      SUB    AL,AL
9502 000030C0 6629D2                    SUB    DX,DX
9503 000030C3 6629C9                    SUB    CX,CX
9504 000030C6 F9                        STC                        ; SET ERROR FLAG
9505 000030C7 EBE7                      JMP    short G5
9506
9507                                ;----------------------------------------
9508                                ;     INITIALIZE DRIVE     (AH = 09H) :
9509                                ;----------------------------------------
9510                                      ; 03/01/2015
9511                                      ; According to ATA-ATAPI specification v2.0 to v5.0
9512                                      ; logical sector per logical track
9513                                      ; and logical heads - 1 would be set but
9514                                      ; it is seen as it will be good
9515                                      ; if physical parameters will be set here
9516                                      ; because, number of heads <= 16.
9517                                      ; (logical heads usually more than 16)
9518                                      ; NOTE: ATA logical parameters (software C, H, S)
9519                                      ;     == INT 13h physical parameters
9520
9521                                ;INIT_DRV:
9522                                ;     MOV    byte [CMD_BLOCK+6],SET_PARM_CMD
9523                                ;     CALL   GET_VEC             ; ES:BX -> PARAMETER BLOCK
9524                                ;     MOV    AL,[ES:BX+2]        ; GET NUMBER OF HEADS
9525                                ;     DEC    AL                  ; CONVERT TO 0-INDEX
```

```
9526                                    ;     MOV   AH,[CMD_BLOCK+5]    ; GET SDH REGISTER
9527                                    ;     AND   AH,0F0H             ; CHANGE HEAD NUMBER
9528                                    ;     OR    AH,AL               ; TO MAX HEAD
9529                                    ;     MOV   [CMD_BLOCK+5],AH
9530                                    ;     MOV   AL,[ES:BX+14]       ; MAX SECTOR NUMBER
9531                                    ;     MOV   [CMD_BLOCK+1],AL
9532                                    ;     SUB   AX,AX
9533                                    ;     MOV   [CMD_BLOCK+3],AL    ; ZERO FLAGS
9534                                    ;     CALL  COMMAND             ; TELL CONTROLLER
9535                                    ;     JNZ   short INIT_EXIT           ; CONTROLLER BUSY ERROR
9536                                    ;     CALL  NOT_BUSY            ; WAIT FOR IT TO BE DONE
9537                                    ;     JNZ   short INIT_EXIT           ; TIME OUT
9538                                    ;     CALL  CHECK_STATUS
9539                                    ;INIT_EXIT:
9540                                    ;     RETn
9541
9542                                    ; 04/01/2015
9543                                    ; 02/01/2015 - Derived from from AWARD BIOS 1999
9544                                    ;                           AHDSK.ASM - INIT_DRIVE
9545                                    INIT_DRV:
9546                                          ;xor   ah,ah
9547 000030C9 31C0                             xor   eax, eax ; 21/02/2015
9548 000030CB B00B                             mov   al,11 ; Physical heads from translated HDPT
9549 000030CD 3825[31480000]                    cmp     [LBAMode], ah   ; 0
9550 000030D3 7702                             ja    short idrv0
9551 000030D5 B002                             mov   al,2  ; Physical heads from standard HDPT
9552                                    idrv0:
9553                                          ; DL = drive number (0 based)
9554 000030D7 E8C7030000                       call  GET_VEC
9555                                          ;push  bx
9556 000030DC 53                               push  ebx ; 21/02/2015
9557                                          ;add   bx,ax
9558 000030DD 01C3                             add   ebx, eax
9559                                          ;; 05/01/2015
9560 000030DF 8A25[BA2C0000]                   mov   ah, [hf_m_s] ; drive number (0= master, 1= slave)
9561                                          ;;and  ah,1
9562 000030E5 C0E404                           shl   ah,4
9563 000030E8 80CCA0                           or    ah,0A0h  ; Drive/Head register - 10100000b (A0h)
9564                                          ;mov   al,[es:bx]
9565 000030EB 8A03                             mov   al, [ebx] ; 21/02/2015
9566 000030ED FEC8                             dec   al      ; last head number
9567                                          ;and   al,0Fh
9568 000030EF 08E0                             or    al,ah   ; lower 4 bits for head number
9569                                          ;
9570 000030F1 C645FE91                         mov   byte [CMD_BLOCK+6],SET_PARM_CMD
9571 000030F5 8845FD                           mov   [CMD_BLOCK+5],al
9572                                          ;pop   bx
9573 000030F8 5B                               pop   ebx
9574 000030F9 29C0                             sub   eax, eax ; 21/02/2015
```

```
9575 000030FB B004                      mov    al,4 ; Physical sec per track from translated HDPT
9576 000030FD 803D[31480000]00          cmp    byte [LBAMode], 0
9577 00003104 7702                      ja     short idrv1
9578 00003106 B00E                      mov    al,14 ; Physical sec per track from standard HDPT
9579                                 idrv1:
9580                                    ;xor   ah,ah
9581                                    ;add   bx,ax
9582 00003108 01C3                      add    ebx, eax ; 21/02/2015
9583                                    ;mov   al,[es:bx]
9584                                                ; sector number
9585 0000310A 8A03                      mov    al, [ebx]
9586 0000310C 8845F9                    mov    [CMD_BLOCK+1],al
9587 0000310F 28C0                      sub    al,al
9588 00003111 8845FB                    mov    [CMD_BLOCK+3],al  ; ZERO FLAGS
9589 00003114 E8C4010000                call   COMMAND        ; TELL CONTROLLER
9590 00003119 750C                      jnz    short INIT_EXIT       ; CONTROLLER BUSY ERROR
9591 0000311B E872020000                call   NOT_BUSY       ; WAIT FOR IT TO BE DONE
9592 00003120 7505                      jnz    short INIT_EXIT       ; TIME OUT
9593 00003122 E8C3020000                call   CHECK_STATUS
9594                                 INIT_EXIT:
9595 00003127 C3                        RETn
9596
9597                                 ;---------------------------------------
9598                                 ;     READ LONG         (AH = 0AH) :
9599                                 ;---------------------------------------
9600
9601                                 RD_LONG:
9602                                    ;MOV   @CMD_BLOCK+6,READ_CMD OR ECC_MODE
9603 00003128 C645FE22                    mov     byte [CMD_BLOCK+6],READ_CMD + ECC_MODE
9604 0000312C E9E0000000                  JMP     COMMANDI
9605
9606                                 ;---------------------------------------
9607                                 ;     WRITE LONG        (AH = 0BH) :
9608                                 ;---------------------------------------
9609
9610                                 WR_LONG:
9611                                    ;MOV   @CMD_BLOCK+6,WRITE_CMD OR ECC_MODE
9612 00003131 C645FE32                    MOV     byte [CMD_BLOCK+6],WRITE_CMD + ECC_MODE
9613 00003135 E92C010000                  JMP     COMMANDO
9614
9615                                 ;---------------------------------------
9616                                 ;     SEEK              (AH = 0CH) :
9617                                 ;---------------------------------------
9618
9619                                 DISK_SEEK:
9620 0000313A C645FE70                    MOV     byte [CMD_BLOCK+6],SEEK_CMD
9621 0000313E E89A010000                CALL   COMMAND
9622 00003143 751C                      JNZ    short DS_EXIT             ; CONTROLLER BUSY ERROR
9623 00003145 E80C020000                CALL   _WAIT
```

```
0000314A 7515                           JNZ     DS_EXIT                 ; TIME OUT ON SEEK
0000314C E899020000                     CALL    CHECK_STATUS
00003151 803D[AF2C0000]40               CMP     byte [DISK_STATUS1],BAD_SEEK
00003158 7507                           JNE     short DS_EXIT
0000315A C605[AF2C0000]00               MOV     byte [DISK_STATUS1],0
                                DS_EXIT:
00003161 C3                             RETn

                                ;---------------------------------------
                                ;       TEST DISK READY      (AH = 10H) :
                                ;---------------------------------------

                                TST_RDY:                        ; WAIT FOR CONTROLLER
00003162 E82B020000                     CALL    NOT_BUSY
00003167 751C                           JNZ     short TR_EX
00003169 8A45FD                         MOV     AL,[CMD_BLOCK+5]    ; SELECT DRIVE
0000316C 668B15[B62C0000]               MOV     DX,[HF_PORT]
00003173 80C206                         add     dl,6
00003176 EE                             OUT     DX,AL
00003177 E886020000                     CALL    CHECK_ST            ; CHECK STATUS ONLY
0000317C 7507                           JNZ     short TR_EX
0000317E C605[AF2C0000]00               MOV     byte [DISK_STATUS1],0     ; WIPE OUT DATA CORRECTED ERROR
                                TR_EX:
00003185 C3                             RETn

                                ;---------------------------------------
                                ;       RECALIBRATE       (AH = 11H) :
                                ;---------------------------------------

                                HDISK_RECAL:
00003186 C645FE10                       MOV     byte [CMD_BLOCK+6],RECAL_CMD ; 10h, 16
0000318A E84E010000                     CALL    COMMAND             ; START THE OPERATION
0000318F 7523                           JNZ     short RECAL_EXIT    ; ERROR
00003191 E8C0010000                     CALL    _WAIT               ; WAIT FOR COMPLETION
00003196 7407                           JZ      short RECAL_X               ; TIME OUT ONE OK ?
00003198 E8B9010000                     CALL    _WAIT               ; WAIT FOR COMPLETION LONGER
0000319D 7515                           JNZ     short RECAL_EXIT    ; TIME OUT TWO TIMES IS ERROR
                                RECAL_X:
0000319F E846020000                     CALL    CHECK_STATUS
000031A4 803D[AF2C0000]40               CMP     byte [DISK_STATUS1],BAD_SEEK ; SEEK NOT COMPLETE
000031AB 7507                           JNE     short RECAL_EXIT    ; IS OK
000031AD C605[AF2C0000]00               MOV     byte [DISK_STATUS1],0
                                RECAL_EXIT:
000031B4 803D[AF2C0000]00               CMP     byte [DISK_STATUS1],0
000031BB C3                             RETn

                                ;---------------------------------------
                                ;      CONTROLLER DIAGNOSTIC (AH = 14H) :
                                ;---------------------------------------
```

```
9673
9674                                    CTLR_DIAGNOSTIC:
9675 000031BC FA                                CLI                             ; DISABLE INTERRUPTS WHILE CHANGING MASK
9676 000031BD E4A1                              IN     AL,INTB01                ; TURN ON SECOND INTERRUPT CHIP
9677                                            ;AND   AL,0BFH
9678 000031BF 243F                              and    al, 3Fh                        ; enable IRQ 14 & IRQ 15
9679                                            ;JMP   $+2
9680                                            IODELAY
9681 000031C1 EB00                      <1>  jmp short $+2
9682 000031C3 EB00                      <1>  jmp short $+2
9683 000031C5 E6A1                              OUT    INTB01,AL
9684                                            IODELAY
9685 000031C7 EB00                      <1>  jmp short $+2
9686 000031C9 EB00                      <1>  jmp short $+2
9687 000031CB E421                              IN     AL,INTA01                ; LET INTERRUPTS PASS THRU TO
9688 000031CD 24FB                              AND    AL,0FBH                  ;  SECOND CHIP
9689                                            ;JMP   $+2
9690                                            IODELAY
9691 000031CF EB00                      <1>  jmp short $+2
9692 000031D1 EB00                      <1>  jmp short $+2
9693 000031D3 E621                              OUT    INTA01,AL
9694 000031D5 FB                                STI
9695 000031D6 E8B7010000                        CALL   NOT_BUSY                 ; WAIT FOR CARD
9696 000031DB 752B                              JNZ    short CD_ERR             ; BAD CARD
9697                                            ;MOV   DX, HF_PORT+7
9698 000031DD 668B15[B62C0000]                  mov    dx, [HF_PORT]
9699 000031E4 80C207                            add    dl, 7
9700 000031E7 B090                              MOV    AL,DIAG_CMD              ; START DIAGNOSE
9701 000031E9 EE                                OUT    DX,AL
9702 000031EA E8A3010000                        CALL   NOT_BUSY                 ; WAIT FOR IT TO COMPLETE
9703 000031EF B480                              MOV    AH,TIME_OUT
9704 000031F1 7517                              JNZ    short CD_EXIT                 ; TIME OUT ON DIAGNOSTIC
9705                                            ;MOV   DX,HF_PORT+1             ; GET ERROR REGISTER
9706 000031F3 668B15[B62C0000]                  mov    dx, [HF_PORT]
9707 000031FA FEC2                              inc    dl
9708 000031FC EC                                IN     AL,DX
9709 000031FD A2[B32C0000]                      MOV    [HF_ERROR],AL            ; SAVE IT
9710 00003202 B400                              MOV    AH,0
9711 00003204 3C01                              CMP    AL,1                     ; CHECK FOR ALL OK
9712 00003206 7402                              JE     SHORT CD_EXIT
9713 00003208 B420                          CD_ERR: MOV  AH,BAD_CNTLR
9714                                        CD_EXIT:
9715 0000320A 8825[AF2C0000]                    MOV    [DISK_STATUS1],AH
9716 00003210 C3                                RETn
9717
9718                                        ;---------------------------------------
9719                                        ; COMMANDI                      :
9720                                        ;     REPEATEDLY INPUTS DATA TILL      :
9721                                        ;     NSECTOR RETURNS ZERO             :
```

```
9722                                    ;---------------------------------------
9723                                    COMMANDI:
9724 00003211 E863020000                        CALL   CHECK_DMA          ; CHECK 64K BOUNDARY ERROR
9725 00003216 724D                              JC     short CMD_ABORT
9726                                            ;MOV   DI,BX
9727 00003218 89DF                              mov    edi, ebx ; 21/02/2015
9728 0000321A E8BE000000                        CALL   COMMAND            ; OUTPUT COMMAND
9729 0000321F 7544                              JNZ    short CMD_ABORT
9730                                    CMD_I1:
9731 00003221 E830010000                        CALL   _WAIT              ; WAIT FOR DATA REQUEST INTERRUPT
9732 00003226 753D                              JNZ    short TM_OUT       ; TIME OUT
9733                                            ;MOV   CX,256             ; SECTOR SIZE IN WORDS
9734 00003228 B900010000                        mov    ecx, 256 ; 21/02/2015
9735                                            ;MOV   DX,HF_PORT
9736 0000322D 668B15[B62C0000]                  mov    dx,[HF_PORT]
9737 00003234 FA                                CLI
9738 00003235 FC                                CLD
9739 00003236 F3666D                            REP    INSW               ; GET THE SECTOR
9740 00003239 FB                                STI
9741 0000323A F645FE02                          TEST   byte [CMD_BLOCK+6],ECC_MODE ; CHECK FOR NORMAL INPUT
9742 0000323E 7419                              JZ     CMD_I3
9743 00003240 E87A010000                        CALL   WAIT_DRQ           ; WAIT FOR DATA REQUEST
9744 00003245 721E                              JC     short TM_OUT
9745                                            ;MOV   DX,HF_PORT
9746 00003247 668B15[B62C0000]                  mov    dx,[HF_PORT]
9747                                            ;MOV   CX,4               ; GET ECC BYTES
9748 0000324E B904000000                        mov    ecx, 4 ; mov cx, 4
9749 00003253 EC                        CMD_I2: IN     AL,DX
9750                                            ;MOV   [ES:DI],AL         ; GO SLOW FOR BOARD
9751 00003254 8807                              mov    [edi], al ; 21/02/2015
9752 00003256 47                                INC    eDI
9753 00003257 E2FA                              LOOP   CMD_I2
9754 00003259 E88C010000                CMD_I3: CALL CHECK_STATUS
9755 0000325E 7505                              JNZ    short CMD_ABORT          ; ERROR RETURNED
9756 00003260 FE4DF9                            DEC    byte [CMD_BLOCK+1]  ; CHECK FOR MORE
9757 00003263 75BC                              JNZ    SHORT CMD_I1
9758                                    CMD_ABORT:
9759 00003265 C3                        TM_OUT: RETn
9760
9761                                    ;---------------------------------------
9762                                    ; COMMANDO                        :
9763                                    ;     REPEATEDLY OUTPUTS DATA TILL     :
9764                                    ;     NSECTOR RETURNS ZERO             :
9765                                    ;---------------------------------------
9766                                    COMMANDO:
9767 00003266 E80E020000                        CALL   CHECK_DMA          ; CHECK 64K BOUNDARY ERROR
9768 0000326B 72F8                              JC     short CMD_ABORT
9769 0000326D 89DE                      CMD_OF: MOV    eSI,eBX ; 21/02/2015
9770 0000326F E869000000                        CALL   COMMAND            ; OUTPUT COMMAND
```

```
9771 00003274 75EF                      JNZ    short CMD_ABORT
9772 00003276 E844010000                CALL   WAIT_DRQ            ; WAIT FOR DATA REQUEST
9773 0000327B 72E8                      JC     short TM_OUT              ; TOO LONG
9774                                CMD_O1: ;PUSH      DS
9775                                        ;PUSH  ES                 ; MOVE ES TO DS
9776                                        ;POP   DS
9777                                        ;MOV   CX,256             ; PUT THE DATA OUT TO THE CARD
9778                                        ;MOV   DX,HF_PORT
9779                                        ; 01/02/2015
9780 0000327D 668B15[B62C0000]              mov    dx, [HF_PORT]
9781                                        ;push  es
9782                                        ;pop   ds
9783                                        ;mov   cx, 256
9784 00003284 B900010000                    mov    ecx, 256 ; 21/02/2015
9785 00003289 FA                            CLI
9786 0000328A FC                            CLD
9787 0000328B F3666F                        REP    OUTSW
9788 0000328E FB                            STI
9789                                        ;POP   DS                 ; RESTORE DS
9790 0000328F F645FE02                      TEST   byte [CMD_BLOCK+6],ECC_MODE ; CHECK FOR NORMAL OUTPUT
9791 00003293 7419                          JZ     short CMD_O3
9792 00003295 E825010000                    CALL   WAIT_DRQ           ; WAIT FOR DATA REQUEST
9793 0000329A 72C9                          JC     short TM_OUT
9794                                        ;MOV   DX,HF_PORT
9795 0000329C 668B15[B62C0000]              mov    dx, [HF_PORT]
9796                                        ;MOV   CX,4               ; OUTPUT THE ECC BYTES
9797 000032A3 B904000000                    mov    ecx, 4  ; mov cx, 4
9798                                CMD_O2: ;MOV AL,[ES:SI]
9799 000032A8 8A06                          mov    al, [esi]
9800 000032AA EE                            OUT    DX,AL
9801 000032AB 46                            INC    eSI
9802 000032AC E2FA                          LOOP   CMD_O2
9803                                CMD_O3:
9804 000032AE E8A3000000                    CALL   _WAIT              ; WAIT FOR SECTOR COMPLETE INTERRUPT
9805 000032B3 75B0                          JNZ    short TM_OUT       ; ERROR RETURNED
9806 000032B5 E830010000                    CALL   CHECK_STATUS
9807 000032BA 75A9                          JNZ    short CMD_ABORT
9808 000032BC F605[B22C0000]08              TEST   byte [HF_STATUS],ST_DRQ   ; CHECK FOR MORE
9809 000032C3 75B8                          JNZ    SHORT CMD_O1
9810                                        ;MOV   DX,HF_PORT+2       ; CHECK RESIDUAL SECTOR COUNT
9811 000032C5 668B15[B62C0000]              mov    dx, [HF_PORT]
9812                                        ;add   dl, 2
9813 000032CC FEC2                          inc    dl
9814 000032CE FEC2                          inc    dl
9815 000032D0 EC                            IN     AL,DX              ;
9816 000032D1 A8FF                          TEST   AL,0FFH            ;
9817 000032D3 7407                          JZ     short CMD_O4              ; COUNT = 0  OK
9818 000032D5 C605[AF2C0000]BB              MOV    byte [DISK_STATUS1],UNDEF_ERR
9819                                                                  ; OPERATION ABORTED - PARTIAL TRANSFER
```

```
                                CMD_O4:
000032DC C3                           RETn
                                
                                ;--------------------------------------------------
                                ; COMMAND                                        :
                                ;     THIS ROUTINE OUTPUTS THE COMMAND BLOCK       :
                                ; OUTPUT                                         :
                                ;     BL = STATUS                                :
                                ;     BH = ERROR REGISTER                        :
                                ;--------------------------------------------------
                                
                                COMMAND:
000032DD 53                           PUSH   eBX                ; WAIT FOR SEEK COMPLETE AND READY
                                      ;;MOV  CX,DELAY_2         ; SET INITIAL DELAY BEFORE TEST
                                COMMAND1:
                                      ;;PUSH CX                 ; SAVE LOOP COUNT
000032DE E87FFEFFFF                   CALL   TST_RDY            ; CHECK DRIVE READY
                                      ;;POP  CX
000032E3 7419                         JZ     short COMMAND2           ; DRIVE IS READY
000032E5 803D[AF2C0000]80               CMP     byte [DISK_STATUS1],TIME_OUT ; TST_RDY TIMED OUT--GIVE UP
                                      ;JZ    short CMD_TIMEOUT
                                      ;;LOOP COMMAND1           ; KEEP TRYING FOR A WHILE
                                      ;JMP   SHORT COMMAND4           ; ITS NOT GOING TO GET READY
000032EC 7507                         jne    short COMMAND4
                                CMD_TIMEOUT:
000032EE C605[AF2C0000]20             MOV    byte [DISK_STATUS1],BAD_CNTLR
                                COMMAND4:
000032F5 5B                           POP    eBX
000032F6 803D[AF2C0000]00               CMP     byte [DISK_STATUS1],0   ; SET CONDITION CODE FOR CALLER
000032FD C3                           RETn
                                COMMAND2:
000032FE 5B                           POP    eBX
000032FF 57                           PUSH   eDI
00003300 C605[B42C0000]00             MOV    byte [HF_INT_FLAG],0      ; RESET INTERRUPT FLAG
00003307 FA                           CLI                      ; INHIBIT INTERRUPTS WHILE CHANGING MASK
00003308 E4A1                         IN     AL,INTB01         ; TURN ON SECOND INTERRUPT CHIP
                                      ;AND   AL,0BFH
0000330A 243F                         and    al, 3Fh                ; Enable IRQ 14 & 15
                                      ;JMP   $+2
                                      IODELAY
0000330C EB00                 <1>  jmp short $+2
0000330E EB00                 <1>  jmp short $+2
00003310 E6A1                         OUT    INTB01,AL
00003312 E421                         IN     AL,INTA01         ; LET INTERRUPTS PASS THRU TO
00003314 24FB                         AND    AL,0FBH           ;  SECOND CHIP
                                      ;JMP   $+2
                                      IODELAY
00003316 EB00                 <1>  jmp short $+2
00003318 EB00                 <1>  jmp short $+2
```

```
9869 0000331A E621                          OUT    INTA01,AL
9870 0000331C FB                            STI
9871 0000331D 31FF                          XOR    eDI,eDI                  ; INDEX THE COMMAND TABLE
9872                                        ;MOV   DX,HF_PORT+1        ; DISK ADDRESS
9873 0000331F 668B15[B62C0000]              mov    dx, [HF_PORT]
9874 00003326 FEC2                          inc    dl
9875 00003328 F605[B12C0000]C0              TEST   byte [CONTROL_BYTE],0C0H ; CHECK FOR RETRY SUPPRESSION
9876 0000332F 7411                          JZ     short COMMAND3
9877 00003331 8A45FE                        MOV    AL, [CMD_BLOCK+6]   ; YES-GET OPERATION CODE
9878 00003334 24F0                          AND    AL,0F0H             ; GET RID OF MODIFIERS
9879 00003336 3C20                          CMP    AL,20H              ; 20H-40H IS READ, WRITE, VERIFY
9880 00003338 7208                          JB     short COMMAND3
9881 0000333A 3C40                          CMP    AL,40H
9882 0000333C 7704                          JA     short COMMAND3
9883 0000333E 804DFE01                      OR     byte [CMD_BLOCK+6],NO_RETRIES
9884                                                                        ; VALID OPERATION FOR RETRY SUPPRESS
9885                                    COMMAND3:
9886 00003342 8A443DF8                      MOV    AL,[CMD_BLOCK+eDI]  ; GET THE COMMAND STRING BYTE
9887 00003346 EE                            OUT    DX,AL               ; GIVE IT TO CONTROLLER
9888                                        IODELAY
9889 00003347 EB00                  <1>  jmp short $+2
9890 00003349 EB00                  <1>  jmp short $+2
9891 0000334B 47                            INC    eDI                 ; NEXT BYTE IN COMMAND BLOCK
9892 0000334C 6642                          INC    DX                  ; NEXT DISK ADAPTER REGISTER
9893 0000334E 6683FF07                      cmp    di, 7  ; 1/1/2015   ; ALL DONE?
9894 00003352 75EE                          JNZ    short COMMAND3           ; NO--GO DO NEXT ONE
9895 00003354 5F                            POP    eDI
9896 00003355 C3                            RETn                        ; ZERO FLAG IS SET
9897
9898                                    ;CMD_TIMEOUT:
9899                                    ;     MOV    byte [DISK_STATUS1],BAD_CNTLR
9900                                    ;COMMAND4:
9901                                    ;     POP    BX
9902                                    ;     CMP    [DISK_STATUS1],0    ; SET CONDITION CODE FOR CALLER
9903                                    ;     RETn
9904
9905                                    ;---------------------------------------
9906                                    ;     WAIT FOR INTERRUPT        :
9907                                    ;---------------------------------------
9908                                    ;WAIT:
9909                                    _WAIT:
9910 00003356 FB                            STI                         ; MAKE SURE INTERRUPTS ARE ON
9911                                        ;SUB   CX,CX                ; SET INITIAL DELAY BEFORE TEST
9912                                        ;CLC
9913                                        ;MOV   AX,9000H             ; DEVICE WAIT INTERRUPT
9914                                        ;INT   15H
9915                                        ;JC    WT2                  ; DEVICE TIMED OUT
9916                                        ;MOV   BL,DELAY_1           ; SET DELAY COUNT
9917
```

```
9918                                    ;mov   bl, WAIT_HDU_INT_HI
9919                                    ;; 21/02/2015
9920                                    ;;mov  bl, WAIT_HDU_INT_HI + 1
9921                                    ;;mov  cx, WAIT_HDU_INT_LO
9922 00003357 B915160500                mov    ecx, WAIT_HDU_INT_LH
9923                                                    ; (AWARD BIOS -> WAIT_FOR_MEM)
9924                                ;-----      WAIT LOOP
9925
9926                                WT1:
9927                                    ;TEST  byte [HF_INT_FLAG],80H    ; TEST FOR INTERRUPT
9928 0000335C F605[B42C0000]C0         test   byte [HF_INT_FLAG],0C0h
9929                                    ;LOOPZ WT1
9930 00003363 7517                     JNZ    short WT3          ; INTERRUPT--LETS GO
9931                                    ;DEC   BL
9932                                    ;JNZ   short WT1          ; KEEP TRYING FOR A WHILE
9933
9934                                WT1_hi:
9935 00003365 E461                     in     al, SYS1 ; 61h (PORT_B)   ; wait for lo to hi
9936 00003367 A810                     test   al, 10h                   ; transition on memory
9937 00003369 75FA                     jnz    short WT1_hi       ; refresh.
9938                                WT1_lo:
9939 0000336B E461                     in     al, SYS1           ; 061h (PORT_B)
9940 0000336D A810                     test   al, 10h
9941 0000336F 74FA                     jz     short WT1_lo
9942 00003371 E2E9                     loop   WT1
9943                                    ;;or   bl, bl
9944                                    ;;jz   short WT2
9945                                    ;;dec  bl
9946                                    ;;jmp  short WT1
9947                                    ;dec   bl
9948                                    ;jnz   short WT1
9949
9950 00003373 C605[AF2C0000]80     WT2:  MOV    byte [DISK_STATUS1],TIME_OUT ; REPORT TIME OUT ERROR
9951 0000337A EB0E                     JMP    SHORT WT4
9952 0000337C C605[AF2C0000]00     WT3:  MOV    byte [DISK_STATUS1],0
9953 00003383 C605[B42C0000]00         MOV    byte [HF_INT_FLAG],0
9954 0000338A 803D[AF2C0000]00     WT4:  CMP    byte [DISK_STATUS1],0     ; SET CONDITION CODE FOR CALLER
9955 00003391 C3                       RETn
9956
9957                                ;--------------------------------------
9958                                ;     WAIT FOR CONTROLLER NOT BUSY     :
9959                                ;--------------------------------------
9960                                NOT_BUSY:
9961 00003392 FB                       STI                        ; MAKE SURE INTERRUPTS ARE ON
9962                                    ;PUSH  eBX
9963                                    ;SUB   CX,CX               ; SET INITIAL DELAY BEFORE TEST
9964 00003393 668B15[B62C0000]         mov    DX, [HF_PORT]
9965 0000339A 80C207                   add    dl, 7               ; Status port (HF_PORT+7)
9966                                    ;MOV   BL,DELAY_1
```

```
                                                        ; wait for 10 seconds
                                ;mov   cx, WAIT_HDU_INT_LO ; 1615h
                                ;;mov  bl, WAIT_HDU_INT_HI ;   05h
                                ;mov   bl, WAIT_HDU_INT_HI + 1
0000339D B915160500             mov    ecx, WAIT_HDU_INT_LH  ; 21/02/2015
                                ;
                        ;;      mov    byte [wait_count], 0    ; Reset wait counter
                        NB1:
000033A2 EC                     IN     AL,DX                   ; CHECK STATUS
                                ;TEST  AL,ST_BUSY
000033A3 2480                   and    al, ST_BUSY
                                ;LOOPNZ      NB1
000033A5 7410                   JZ     short NB2               ; NOT BUSY--LETS GO
                                ;DEC   BL
                                ;JNZ   short NB1               ; KEEP TRYING FOR A WHILE

000033A7 E461                   NB1_hi: IN   AL,SYS1                   ; wait for hi to lo
000033A9 A810                   TEST   AL,010H                         ; transition on memory
000033AB 75FA                   JNZ    SHORT NB1_hi            ; refresh.
000033AD E461                   NB1_lo: IN   AL,SYS1
000033AF A810                   TEST   AL,010H
000033B1 74FA                   JZ     short NB1_lo
000033B3 E2ED                   LOOP   NB1
                                ;dec   bl
                                ;jnz   short NB1
                                ;
                        ;;      cmp    byte [wait_count], 182  ; 10 seconds (182 timer ticks)
                        ;;    jb     short NB1
                                ;
                                ;MOV   [DISK_STATUS1],TIME_OUT   ; REPORT TIME OUT ERROR
                                ;JMP   SHORT NB3
000033B5 B080                   mov    al, TIME_OUT
                        NB2:
                                ;MOV   byte [DISK_STATUS1],0
                        ;NB3:
                                ;POP   eBX
000033B7 A2[AF2C0000]           mov    [DISK_STATUS1], al  ;;; will be set after return
                                ;CMP   byte [DISK_STATUS1],0     ; SET CONDITION CODE FOR CALLER
000033BC 08C0                   or     al, al                  ; (zf = 0 --> timeout)
000033BE C3                     RETn

                        ;---------------------------------------
                        ;     WAIT FOR DATA REQUEST             :
                        ;---------------------------------------
                        WAIT_DRQ:
                                ;MOV   CX,DELAY_3
                                ;MOV   DX,HF_PORT+7
000033BF 668B15[B62C0000]       mov    dx, [HF_PORT]
000033C6 80C207                 add    dl, 7
```

```
10016                                       ;;MOV  bl, WAIT_HDU_DRQ_HI ; 0
10017                                       ;MOV   cx, WAIT_HDU_DRQ_LO ; 1000 (30 milli seconds)
10018                                                                  ; (but it is written as 2000
10019                                                                  ; micro seconds in ATORGS.ASM file
10020                                                                  ; of Award Bios - 1999, D1A0622)
10021 000033C9 B9E8030000                  mov    ecx, WAIT_HDU_DRQ_LH ; 21/02/2015
10022 000033CE EC                    WQ_1: IN     AL,DX                ; GET STATUS
10023 000033CF A808                        TEST   AL,ST_DRQ            ; WAIT FOR DRQ
10024 000033D1 7516                        JNZ    short WQ_OK
10025                                      ;LOOP  WQ_1                 ; KEEP TRYING FOR A SHORT WHILE
10026                                 WQ_hi:
10027 000033D3 E461                        IN     AL,SYS1                    ; wait for hi to lo
10028 000033D5 A810                        TEST   AL,010H                    ; transition on memory
10029 000033D7 75FA                        JNZ    SHORT WQ_hi          ; refresh.
10030 000033D9 E461                  WQ_lo:  IN      AL,SYS1
10031 000033DB A810                        TEST   AL,010H
10032 000033DD 74FA                        JZ     SHORT WQ_lo
10033 000033DF E2ED                        LOOP   WQ_1
10034
10035 000033E1 C605[AF2C0000]80              MOV     byte [DISK_STATUS1],TIME_OUT ; ERROR
10036 000033E8 F9                          STC
10037                                 WQ_OK:
10038 000033E9 C3                          RETn
10039                                 ;WQ_OK:     ;CLC
10040                                 ;     RETn
10041
10042                                 ;---------------------------------------
10043                                 ;     CHECK FIXED DISK STATUS   :
10044                                 ;---------------------------------------
10045                                 CHECK_STATUS:
10046 000033EA E813000000                  CALL   CHECK_ST             ; CHECK THE STATUS BYTE
10047 000033EF 7509                        JNZ    short CHECK_S1             ; AN ERROR WAS FOUND
10048 000033F1 A801                        TEST   AL,ST_ERROR          ; WERE THERE ANY OTHER ERRORS
10049 000033F3 7405                        JZ     short CHECK_S1             ; NO ERROR REPORTED
10050 000033F5 E847000000                  CALL   CHECK_ER             ; ERROR REPORTED
10051                                 CHECK_S1:
10052 000033FA 803D[AF2C0000]00            CMP    byte [DISK_STATUS1],0      ; SET STATUS FOR CALLER
10053 00003401 C3                          RETn
10054
10055                                 ;---------------------------------------
10056                                 ;     CHECK FIXED DISK STATUS BYTE     :
10057                                 ;---------------------------------------
10058                                 CHECK_ST:
10059                                      ;MOV   DX,HF_PORT+7         ; GET THE STATUS
10060 00003402 668B15[B62C0000]            mov    dx, [HF_PORT]
10061 00003409 80C207                      add    dl, 7
10062 0000340C EC                          IN     AL,DX
10063 0000340D A2[B22C0000]                MOV    [HF_STATUS],AL
10064 00003412 B400                        MOV    AH,0
```

```
10065 00003414 A880                     TEST   AL,ST_BUSY          ; IF STILL BUSY
10066 00003416 751A                     JNZ    short CKST_EXIT           ;  REPORT OK
10067 00003418 B4CC                     MOV    AH,WRITE_FAULT
10068 0000341A A820                     TEST   AL,ST_WRT_FLT       ; CHECK FOR WRITE FAULT
10069 0000341C 7514                     JNZ    short CKST_EXIT
10070 0000341E B4AA                     MOV    AH,NOT_RDY
10071 00003420 A840                     TEST   AL,ST_READY         ; CHECK FOR NOT READY
10072 00003422 740E                     JZ     short CKST_EXIT
10073 00003424 B440                     MOV    AH,BAD_SEEK
10074 00003426 A810                     TEST   AL,ST_SEEK_COMPL    ; CHECK FOR SEEK NOT COMPLETE
10075 00003428 7408                     JZ     short CKST_EXIT
10076 0000342A B411                     MOV    AH,DATA_CORRECTED
10077 0000342C A804                     TEST   AL,ST_CORRCTD       ; CHECK FOR CORRECTED ECC
10078 0000342E 7502                     JNZ    short CKST_EXIT
10079 00003430 B400                     MOV    AH,0
10080                           CKST_EXIT:
10081 00003432 8825[AF2C0000]           MOV    [DISK_STATUS1],AH   ; SET ERROR FLAG
10082 00003438 80FC11                   CMP    AH,DATA_CORRECTED   ; KEEP GOING WITH DATA CORRECTED
10083 0000343B 7403                     JZ     short CKST_EX1
10084 0000343D 80FC00                   CMP    AH,0
10085                           CKST_EX1:
10086 00003440 C3                       RETn
10087
10088                           ;---------------------------------------
10089                           ;     CHECK FIXED DISK ERROR REGISTER :
10090                           ;---------------------------------------
10091                           CHECK_ER:
10092                                   ;MOV   DX, HF_PORT+1       ; GET THE ERROR REGISTER
10093 00003441 668B15[B62C0000]         mov    dx, [HF_PORT]       ;
10094 00003448 FEC2                     inc    dl
10095 0000344A EC                       IN     AL,DX
10096 0000344B A2[B32C0000]             MOV    [HF_ERROR],AL
10097 00003450 53                       PUSH   eBX ; 21/02/2015
10098 00003451 B908000000               MOV    eCX,8               ; TEST ALL 8 BITS
10099 00003456 D0E0                     SHL    AL,1                ; MOVE NEXT ERROR BIT TO CARRY
10100 00003458 7202                     JC     short CK2           ; FOUND THE ERROR
10101 0000345A E2FA                     LOOP   CK1                 ; KEEP TRYING
10102 0000345C BB[70340000]             MOV    eBX, ERR_TBL        ; COMPUTE ADDRESS OF
10103 00003461 01CB                     ADD    eBX,eCX                  ; ERROR CODE
10104                                   ;;MOV  AH,BYTE [CS:BX]          ; GET ERROR CODE
10105                                   ;mov   ah, byte [bx]
10106 00003463 8A23                     mov    ah, byte [ebx] ; 21/02/2015
10107 00003465 8825[AF2C0000]           MOV    [DISK_STATUS1],AH   ; SAVE ERROR CODE
10108 0000346B 5B                       POP    eBX
10109 0000346C 80FC00                   CMP    AH,0
10110 0000346F C3                       RETn
10111                           ERR_TBL:
10112 00003470 E0                       DB     NO_ERR
10113 00003471 024001BB                 DB     BAD_ADDR_MARK,BAD_SEEK,BAD_CMD,UNDEF_ERR
```

```
10114 00003475 04BB100A                     DB    RECORD_NOT_FND,UNDEF_ERR,BAD_ECC,BAD_SECTOR
10115
10116                                 ;-----------------------------------------------------
10117                                 ; CHECK_DMA                                         :
10118                                 ;  -CHECK ES:BX AND # SECTORS TO MAKE SURE THAT IT WILL :
10119                                 ;   FIT WITHOUT SEGMENT OVERFLOW.                    :
10120                                 ;  -ES:BX HAS BEEN REVISED TO THE FORMAT SSSS:000X  :
10121                                 ;  -OK IF # SECTORS < 80H (7FH IF LONG READ OR WRITE)     :
10122                                 ;  -OK IF # SECTORS = 80H (7FH) AND BX <= 00H (04H) :
10123                                 ;  -ERROR OTHERWISE                                 :
10124                                 ;-----------------------------------------------------
10125                                 CHECK_DMA:
10126 00003479 6650                         PUSH   AX                  ; SAVE REGISTERS
10127 0000347B 66B80080                     MOV    AX,8000H            ; AH = MAX # SECTORS AL = MAX OFFSET
10128 0000347F F645FE02                       TEST    byte [CMD_BLOCK+6],ECC_MODE
10129 00003483 7404                         JZ     short CKD1
10130 00003485 66B8047F                     MOV    AX,7F04H            ; ECC IS 4 MORE BYTES
10131 00003489 3A65F9                 CKD1: CMP    AH, [CMD_BLOCK+1]   ; NUMBER OF SECTORS
10132 0000348C 7706                         JA     short CKDOK         ; IT WILL FIT
10133 0000348E 7208                         JB     short CKDERR        ; TOO MANY
10134 00003490 38D8                         CMP    AL,BL               ; CHECK OFFSET ON MAX SECTORS
10135 00003492 7204                         JB     short CKDERR        ; ERROR
10136 00003494 F8                     CKDOK:       CLC                       ; CLEAR CARRY
10137 00003495 6658                         POP    AX
10138 00003497 C3                           RETn                       ; NORMAL RETURN
10139 00003498 F9                     CKDERR: STC                      ; INDICATE ERROR
10140 00003499 C605[AF2C0000]09               MOV     byte [DISK_STATUS1],DMA_BOUNDARY
10141 000034A0 6658                         POP    AX
10142 000034A2 C3                           RETn
10143
10144                                 ;----------------------------------------
10145                                 ;     SET UP ES:BX-> DISK PARMS :
10146                                 ;----------------------------------------
10147
10148                                 ; INPUT -> DL = 0 based drive number
10149                                 ; OUTPUT -> ES:BX = disk parameter table address
10150
10151                                 GET_VEC:
10152                                       ;SUB   AX,AX               ; GET DISK PARAMETER ADDRESS
10153                                       ;MOV   ES,AX
10154                                       ;TEST  DL,1
10155                                       ;JZ    short GV_0
10156                                 ;     LES    BX,[HF1_TBL_VEC]    ; ES:BX -> DRIVE PARAMETERS
10157                                 ;     JMP    SHORT GV_EXIT
10158                                 ;GV_0:
10159                                 ;     LES    BX,[HF_TBL_VEC]           ; ES:BX -> DRIVE PARAMETERS
10160                                 ;
10161                                       ;xor   bh, bh
10162 000034A3 31DB                        xor    ebx, ebx
```

```
10163 000034A5 88D3                     mov    bl, dl
10164                                   ;;02/01/2015
10165                                   ;;shl  bl, 1               ; port address offset
10166                                   ;;mov  ax, [bx+hd_ports]   ; Base port address (1F0h, 170h)
10167                                   ;;shl  bl, 1               ; dpt pointer offset
10168 000034A7 C0E302                   shl    bl, 2  ;;
10169                                   ;add   bx, HF_TBL_VEC            ; Disk parameter table pointer
10170 000034AA 81C3[BC2C0000]           add    ebx, HF_TBL_VEC ; 21/02/2015
10171                                   ;push  word [bx+2]         ; dpt segment
10172                                   ;pop   es
10173                                   ;mov   bx, [bx]            ; dpt offset
10174 000034B0 8B1B                     mov    ebx, [ebx]
10175                                 ;GV_EXIT:
10176 000034B2 C3                       RETn
10177
10178                                 hdc1_int: ; 21/02/2015
10179                                 ;--- HARDWARE INT 76H -- ( IRQ LEVEL  14 ) ----------------------
10180                                 ;                                                       :
10181                                 ;     FIXED DISK INTERRUPT ROUTINE                            :
10182                                 ;                                                       :
10183                                 ;-------------------------------------------------------------
10184
10185                                 ; 22/12/2014
10186                                 ; IBM PC-XT Model 286 System BIOS Source Code - DISK.ASM (HD_INT)
10187                                 ;      '11/15/85'
10188                                 ; AWARD BIOS 1999 (D1A0622)
10189                                 ;     Source Code - ATORGS.ASM (INT_HDISK, INT_HDISK1)
10190
10191                                 ;int_76h:
10192                                 HD_INT:
10193 000034B3 6650                       PUSH   AX
10194 000034B5 1E                         PUSH   DS
10195                                     ;CALL  DDS
10196                                     ; 21/02/2015 (32 bit, 386 pm modification)
10197 000034B6 66B81000                   mov    ax, KDATA
10198 000034BA 8ED8                       mov    ds, ax
10199                                     ;
10200                                     ;;MOV  @HF_INT_FLAG,0FFH   ; ALL DONE
10201                                       ;mov     byte [CS:HF_INT_FLAG], 0FFh
10202 000034BC C605[B42C0000]FF           mov    byte [HF_INT_FLAG], 0FFh
10203                                     ;
10204 000034C3 6652                       push   dx
10205 000034C5 66BAF701                   mov    dx, HDC1_BASEPORT+7 ; Status Register (1F7h)
10206                                                                ; Clear Controller
10207                                 Clear_IRQ1415:                       ; (Award BIOS - 1999)
10208 000034C9 EC                         in     al, dx              ;
10209 000034CA 665A                       pop    dx
10210                                     NEWIODELAY
10211 000034CC E6EB                 <1>  out 0ebh,al
```

```
10212                                           ;
10213 000034CE B020                             MOV    AL,EOI             ; NON-SPECIFIC END OF INTERRUPT
10214 000034D0 E6A0                             OUT    INTB00,AL          ; FOR CONTROLLER #2
10215                                           ;JMP   $+2                ; WAIT
10216                                           NEWIODELAY
10217 000034D2 E6EB                     <1>  out 0ebh,al
10218 000034D4 E620                             OUT    INTA00,AL          ; FOR CONTROLLER #1
10219 000034D6 1F                               POP    DS
10220                                           ;STI                      ; RE-ENABLE INTERRUPTS
10221                                           ;MOV   AX,9100H           ; DEVICE POST
10222                                           ;INT   15H                ;  INTERRUPT
10223                                       irq15_iret: ; 25/02/2015
10224 000034D7 6658                             POP    AX
10225 000034D9 CF                               IRETd                     ; RETURN FROM INTERRUPT
10226
10227                                       hdc2_int: ; 21/02/2015
10228                                       ;++++ HARDWARE INT 77H ++ ( IRQ LEVEL  15 ) +++++++++++++++++++++
10229                                       ;                                                       :
10230                                       ;     FIXED DISK INTERRUPT ROUTINE                            :
10231                                       ;                                                       :
10232                                       ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
10233
10234                                       ;int_77h:
10235                                       HD1_INT:
10236 000034DA 6650                             PUSH   AX
10237                                           ; Check if that is a spurious IRQ (from slave PIC)
10238                                           ; 25/02/2015 (source: http://wiki.osdev.org/8259_PIC)
10239 000034DC B00B                             mov    al, 0Bh  ; In-Service Register
10240 000034DE E6A0                             out    0A0h, al
10241 000034E0 EB00                               jmp short $+2
10242 000034E2 EB00                             jmp short $+2
10243 000034E4 E4A0                             in     al, 0A0h
10244 000034E6 2480                             and    al, 80h ; bit 7 (is it real IRQ 15 or fake?)
10245 000034E8 74ED                             jz     short irq15_iret ; Fake (spurious)IRQ, do not send EOI)
10246                                           ;
10247 000034EA 1E                               PUSH   DS
10248                                           ;CALL  DDS
10249                                           ; 21/02/2015 (32 bit, 386 pm modification)
10250 000034EB 66B81000                         mov    ax, KDATA
10251 000034EF 8ED8                             mov    ds, ax
10252                                           ;
10253                                           ;;MOV  @HF_INT_FLAG,0FFH   ; ALL DONE
10254                                             ;or      byte [CS:HF_INT_FLAG],0C0h
10255 000034F1 800D[B42C0000]C0                 or     byte [HF_INT_FLAG], 0C0h
10256                                           ;
10257 000034F8 6652                             push   dx
10258 000034FA 66BA7701                         mov    dx, HDC2_BASEPORT+7 ; Status Register (177h)
10259                                                                      ; Clear Controller (Award BIOS 1999)
10260 000034FE EBC9                             jmp    short Clear_IRQ1415
```

```
10261
10262
10263                                   ;////////////////////////////////////////////////////////////
10264                                   ;; END OF DISK I/O ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
10265
10266
10267                                   ; Beginning of DSECTPM (Display Disk Sectors in Protected Mode) code
10268                                   ;; ////////////////////////////////////////////////////////
10269                                   ;, 12/02/2015
10270                                   ;; Display Disk Sectors in protected mode (12/02/2015, unix386.s)
10271                                   ;; Retro UNIX 386 v1 - DISK I/O Test
10272
10273
10274                                   ; display disk sector data [ Retro Unix 386 v1 - test ]
10275                                   ; modified from 'DSECTRM2.S' (02/02/2015)
10276                                   ; by Erdogan Tan
10277                                   ;
10278
10279                                   ;; 22/02/2015
10280                                   ;, Enable/Setup Real Time Clock Interrupt (dsectpm.s)
10281                                   ;;
10282                                   setup_rtc_int:
10283                                   ; source: http://wiki.osdev.org/RTC
10284 00003500 FA                            cli          ; disable interrupts
10285                                         ; default int frequency is 1024 Hz (Lower 4 bits of register A is 0110b or 6)
10286                                         ; in order to change this ...
10287                                         ; frequency  = 32768 >> (rate-1) --> 32768 >> 5 = 1024
10288                                         ; (rate must be above 2 and not over 15)
10289                                         ; new rate = 15 --> 32768 >> (15-1) = 2 Hz
10290 00003501 B08A                          mov    al, 8Ah
10291 00003503 E670                          out    70h, al ; set index to register A, disable NMI
10292 00003505 90                            nop
10293 00003506 E471                          in     al, 71h ; get initial value of register A
10294 00003508 88C4                          mov    ah, al
10295 0000350A 80E4F0                        and    ah, 0F0h
10296 0000350D B08A                          mov    al, 8Ah
10297 0000350F E670                          out    70h, al ; reset index to register A
10298 00003511 88E0                          mov    al, ah
10299 00003513 0C0F                          or     al, 0Fh      ; new rate (0Fh -> 15)
10300 00003515 E671                          out    71h, al ; write only our rate to A. Note, rate is the bottom 4 bits.
10301                                         ; enable RTC interrupt
10302 00003517 B08B                          mov    al, 8Bh ;
10303 00003519 E670                          out    70h, al ; select register B and disable NMI
10304 0000351B 90                            nop
10305 0000351C E471                          in     al, 71h ; read the current value of register B
10306 0000351E 88C4                          mov    ah, al  ;
10307 00003520 B08B                          mov    al, 8Bh ;
10308 00003522 E670                          out    70h, al ; set the index again (a read will reset the index to register
B)
```

```
 10309 00003524 88E0                    mov    al, ah  ;
 10310 00003526 0C40                    or     al, 40h ;
 10311 00003528 E671                    out    71h, al ; write the previous value ORed with 0x40. This turns on bit 6
of register B
 10312 0000352A FB                     sti
 10313 0000352B C3                     retn
 10314
 10315                          ; 25/11/2014
 10316
 10317                          ESCKey       equ 1Bh    ;27
 10318                          ENTERKey equ 0Dh    ;13
 10319                          SPACEKey equ 20h    ;32
 10320                          BACKSPC      equ 08h    ; 8
 10321                          DELKey       equ 53E0h
 10322                          F1Key  equ 3B00h
 10323                          F2Key  equ 3C00h
 10324                          F3Key  equ 3D00h ; 19/12/2014
 10325                          HOMEKey  equ 47E0h
 10326                          ENDKey       equ 4FE0h
 10327                          PgUpKey      equ 49E0h
 10328                          PgDnKey  equ 51E0h
 10329
 10330                          ; 30/12/2014 (0B000h -> 9000h)
 10331                          ; 10/12/2014
 10332                          ;DPT_SEGM equ 9000h  ; FDPT segment (EDD v1.1, EDD v3)
 10333                          ;
 10334                          ;HD0_DPT  equ 0        ; Disk parameter table address for hd0
 10335                          ;HD1_DPT  equ 32    ; Disk parameter table address for hd1
 10336                          ;HD2_DPT  equ 64    ; Disk parameter table address for hd2
 10337                          ;HD3_DPT  equ 96    ; Disk parameter table address for hd3
 10338
 10339                          ; FDPT (Phoenix, Enhanced Disk Drive Specification v1.1, v3.0)
 10340                          ;      (HDPT: Programmer's Guide to the AMIBIOS, 1993)
 10341                          ;
 10342                          ;FDPT_CYLS equ 0  ; 1 word, number of cylinders
 10343                          ;FDPT_HDS  equ 2  ; 1 byte, number of heads
 10344                          ;FDPT_TT      equ 3  ; 1 byte, A0h = translated FDPT with logical values
 10345                                      ; otherwise it is standard FDPT with physical values
 10346                          ;FDPT_PCMP equ 5  ; 1 word, starting write precompensation cylinder
 10347                                      ; (obsolete for IDE/ATA drives)
 10348                          ;FDPT_CB   equ 8  ; 1 byte, drive control byte
 10349                                       ; Bits 7-6 : Enable or disable retries (00h = enable)
 10350                                       ; Bit 5    : 1 = Defect map is located at last cyl. + 1
 10351                                       ; Bit 4 : Reserved. Always 0
 10352                                       ; Bit 3 : Set to 1 if more than 8 heads
 10353                                       ; Bit 2-0 : Reserved. Alsways 0
 10354                          ;FDPT_LZ   equ 12 ; 1 word, landing zone (obsolete for IDE/ATA drives)
 10355                          ;FDPT_SPT  equ 14 ; 1 byte, sectors per track
 10356
```

```
10357                                  dsectpm: ; 16/02/2015
10358                                        ;mov  word [ttychr], 0
10359                                  display_sectors:
10360                                        ; 20/02/2015
10361                                        ; 12/02/2015 (protected mode modification, unix386.s)
10362                                        ; 26/11/2014 - 04/12/2014 (dsectrm2.s)
10363                                        ;
10364 0000352C E80F060000                    call  hide_cursor
10365                                        ; Save video page (before displaying sector)
10366
10367 00003531 BE00800B00                    mov   esi, 0B8000h
10368 00003536 89F7                          mov   edi, esi
10369 00003538 81C7C05D0000                  add   edi, 5DC0h  ; 4000*6 (video page 6)
10370 0000353E B9E8030000                    mov   ecx, 1000
10371 00003543 F3A5                          rep   movsd
10372
10373                                        ; Save cursor position
10374                                        ;mov  ax, [cursor_posn] ; cursor pos.
10375                                                               ; for video page 0.
10376                                        ;mov  [cursor_posb], ax
10377 00003545 E8E6040000                    call  clear_frame
10378 0000354A EB35                          jmp   short dscl_0
10379                                  dscl_esc:
10380 0000354C E8A0030000                    call  restore_video_page
10381                                  dscl_getc:
10382 00003551 E87F030000                    call  getch
10383                                        ;
10384 00003556 3C1B                          cmp   al, ESCKey
10385 00003558 0F8441030000                    je      dscl_exit
10386 0000355E C605[F2460000]00              mov   byte [dscmd], 0
10387 00003565 663D003B                      cmp   ax, F1Key
10388 00003569 7416                            je  short dscl_0
10389                                        ; 19/12/2014
10390 0000356B FE05[F2460000]                inc   byte [dscmd]
10391 00003571 663D003D                      cmp   ax, F3Key
10392 00003575 0F8573020000                    jne     dscl_5
10393 0000357B FE05[F2460000]                inc   byte [dscmd] ; Display disk params.
10394                                  dscl_0:
10395 00003581 E856030000                    call  save_video_page
10396 00003586 BE[DE470000]                  mov   esi, F1_ib ; F1 (Change drive)
10397                                                         ; Inputbox address
10398                                  dscl_ib:
10399 0000358B E800050000                    call  inputbox
10400                                              ; cursor position in DX
10401 00003590 E8AF050000                    call  show_cursor
10402                                              ; cursor blinks at current position
10403                                  dscl_3:
10404 00003595 E83B030000                    call  getch
10405 0000359A 3C1B                          cmp   al, ESCKey
```

```
10406 0000359C 7507                     jne    short dscl_27
10407 0000359E E89D050000                 call    hide_cursor
10408 000035A3 EBA7                     jmp    short dscl_esc
10409                                 dscl_27:
10410 000035A5 3C20                     cmp    al, SPACEKey
10411 000035A7 7438                     je     short dscl_4
10412 000035A9 3C0D                     cmp    al, ENTERKey
10413 000035AB 7434                     je     short dscl_4
10414                                   ;
10415 000035AD 31DB                     xor    ebx, ebx ; 20/02/2015
10416 000035AF 803D[F2460000]01         cmp     byte [dscmd], 1
10417 000035B6 7466                     je     short dscl_12
10418                                   ;
10419 000035B8 3C30                     cmp    al, '0'
10420 000035BA 72D9                     jb     short dscl_3
10421 000035BC 3C35                     cmp    al, '5'
10422 000035BE 77D5                     ja     short dscl_3
10423 000035C0 8B3D[ED460000]           mov    edi, [current_txtpos]
10424 000035C6 AA                       stosb
10425 000035C7 2C30                     sub    al, '0'
10426 000035C9 88C2                     mov    dl, al
10427 000035CB 30F6                     xor    dh, dh
10428 000035CD 88C3                     mov    bl, al
10429 000035CF C0E302                   shl    bl, 2  ; *4
10430 000035D2 81C3[9B460000]           add    ebx, ds_sec ; current_sector
10431 000035D8 8B0B                     mov    ecx, [ebx]
10432 000035DA BE[D04A0000]             mov    esi, buffer
10433 000035DF EBB4                     jmp    short dscl_3
10434                                 dscl_4:
10435 000035E1 803D[98460000]00         cmp    byte [inds],  0 ; display other half or not ?
10436 000035E8 0F8717010000               ja      dscl_oh         ; other half
10437 000035EE 51                       push   ecx
10438                                   ; 17/02/2015
10439                                   ; save regs (ESI, EAX, DX)
10440 000035EF 6652                     push   dx
10441 000035F1 E84A050000               call   hide_cursor
10442                                   ; restore regs (ESI, EAX, DX)
10443 000035F6 665A                     pop    dx
10444 000035F8 58                       pop    eax
10445                                     ; 19/12/2014
10446 000035F9 803D[F2460000]01         cmp     byte [dscmd], 1 ; Requested function ?
10447 00003600 0F84C2000000               je      dscl_17         ; Change sector (F2)
10448 00003606 0F8234010000               jb      dscl_ns         ; Change drive (F1)
10449                                   ; Display disk parameters (dscmd = 2)
10450                                   ; 31/12/2014
10451 0000360C 80FA02                   cmp    dl, 2
10452 0000360F 7203                     jb     short dscl_28
10453 00003611 80C27E                   add    dl, 7Eh
10454                                 dscl_28:
```

```
10455 00003614 E872060000                call  dskprm
10456 00003619 E92EFFFFFF                  jmp     dscl_esc
10457                                dscl_12:
10458 0000361E 663DE053                    cmp   ax, DELKey   ; DEL key
10459 00003622 7404                        je    short dscl_bs
10460 00003624 3C08                        cmp   al, BACKSPC  ; Backspace key
10461 00003626 7536                        jne   short dscl_13
10462                                dscl_bs:
10463 00003628 803D[F1460000]00            cmp   byte [txtposoff], 0
10464 0000362F 0F8660FFFFFF                  jna     dscl_3
10465 00003635 FE0D[F1460000]              dec   byte [txtposoff]
10466 0000363B FE0D[3C490000]              dec   byte [cursor_posn]
10467                                      ;mov  dx, [cursor_posn]
10468 00003641 E8CC110000                  call  set_cposn
10469 00003646 31DB                        xor   ebx, ebx
10470 00003648 8A1D[F1460000]              mov   bl, [txtposoff]
10471 0000364E FE0D[F1460000]              dec   byte [txtposoff]
10472 00003654 FE0D[3C490000]              dec   byte [cursor_posn]
10473 0000365A B020                        mov   al, 20h
10474 0000365C EB1B                          jmp     short dscl_14
10475                                dscl_13:
10476 0000365E 8A1D[F1460000]              mov   bl, [txtposoff]
10477 00003664 80FB08                      cmp   bl, 8
10478 00003667 0F8328FFFFFF                  jnb     dscl_3
10479                                      ;
10480 0000366D 3C30                        cmp   al, '0'
10481 0000366F 0F8220FFFFFF                  jb      dscl_3
10482 00003675 3C39                        cmp   al, '9'
10483 00003677 772F                        ja    short dscl_15
10484                                dscl_14:
10485 00003679 D0E3                        shl   bl, 1
10486 0000367B 8B35[ED460000]              mov   esi, [current_txtpos]
10487 00003681 01F3                        add   ebx, esi
10488 00003683 8803                        mov   [ebx], al
10489                                      ;
10490 00003685 803D[F1460000]08            cmp   byte [txtposoff], 8
10491 0000368C 0F8D03FFFFFF                jge   dscl_3 ; JGE !
10492 00003692 FE05[F1460000]              inc   byte [txtposoff]
10493 00003698 FE05[3C490000]              inc   byte [cursor_posn]
10494                                      ;mov  dx, [cursor_posn]
10495 0000369E E86F110000                  call  set_cposn
10496 000036A3 E9EDFEFFFF                    jmp     dscl_3
10497                                dscl_15:
10498 000036A8 3C41                        cmp   al, 'A'
10499 000036AA 0F82E5FEFFFF                  jb      dscl_3
10500 000036B0 3C46                        cmp   al, 'F'
10501 000036B2 76C5                          jna     short dscl_14
10502                                dscl_16:
10503 000036B4 3C61                        cmp   al, 'a'
```

```
10504 000036B6 0F82D9FEFFFF                jb      dscl_3
10505 000036BC 3C66                       cmp   al, 'f'
10506 000036BE 0F87D1FEFFFF                ja      dscl_3
10507 000036C4 2C20                       sub   al, 'a' - 'A'
10508 000036C6 EBB1                       jmp   short dscl_14
10509                                     ;
10510                                   dscl_17:
10511 000036C8 8B35[ED460000]                mov   esi, [current_txtpos]
10512 000036CE 31C0                          xor   eax, eax
10513 000036D0 A2[F1460000]                  mov   byte [txtposoff], al ; 0
10514 000036D5 50                            push  eax  ; sector value (reset)
10515                                   dscl_18:
10516 000036D6 66AD                          lodsw
10517 000036D8 3C30                          cmp   al, '0'
10518 000036DA 7219                          jb    short dscl_22
10519                                   dscl_19:
10520 000036DC 29C9                          sub   ecx, ecx
10521 000036DE BB[88460000]                  mov   ebx, hexchrs
10522                                   dscl_20:
10523 000036E3 3A03                          cmp   al, [ebx]
10524 000036E5 7405                          je    short dscl_21
10525                                        ;cmp  cl, 15
10526                                        ;jnb  short dscl_22
10527 000036E7 FEC1                          inc   cl
10528 000036E9 43                            inc   ebx
10529 000036EA EBF7                          jmp   short dscl_20
10530                                   dscl_21:
10531 000036EC 58                            pop   eax
10532 000036ED C1E004                        shl   eax, 4 ; * 16
10533 000036F0 01C8                          add   eax, ecx
10534 000036F2 50                            push  eax
10535 000036F3 EBE1                          jmp   short dscl_18
10536                                   dscl_22:
10537 000036F5 8A15[99460000]                mov   dl, [ds_drv]
10538 000036FB 30F6                          xor   dh, dh
10539 000036FD 58                            pop   eax
10540 000036FE BE[D04A0000]                  mov   esi, buffer
10541 00003703 EB3B                         jmp     short dscl_ns
10542                                   dscl_oh:
10543 00003705 8A15[99460000]                mov   dl, [ds_drv]
10544 0000370B 31DB                          xor   ebx, ebx
10545 0000370D 88D3                          mov   bl, dl
10546 0000370F C0E302                        shl   bl, 2
10547 00003712 81C3[9B460000]                add   ebx, ds_sec
10548 00003718 8B03                          mov   eax, [ebx]
10549 0000371A BE[D04A0000]                  mov   esi, buffer
10550                                        ;
10551 0000371F 8A35[9A460000]                mov   dh, [ds_drv+1]
10552 00003725 08F6                          or    dh, dh
```

```
10553 00003727 7404                     jz     short dscl_nh ; second half of sector (0->1)
10554 00003729 30F6                     xor    dh, dh       ; reset (0)
10555 0000372B EB08                     jmp    short dscl_nx
10556                                 dscl_nh:
10557 0000372D 81C600010000             add    esi, 256
10558 00003733 FEC6                     inc    dh
10559                                 dscl_nx:
10560 00003735 8835[9A460000]           mov    [ds_drv+1], dh
10561 0000373B E98D000000                 jmp     dscl_25
10562                                 dscl_ns:
10563 00003740 8835[9A460000]           mov    [ds_drv+1], dh
10564 00003746 31DB                     xor    ebx, ebx
10565 00003748 88D3                     mov    bl, dl
10566 0000374A C0E302                   shl    bl, 2
10567 0000374D 81C3[9B460000]           add    ebx, ds_sec
10568 00003753 3A15[99460000]           cmp    dl, [ds_drv]
10569 00003759 7504                     jne    short dscl_23
10570 0000375B 3B03                     cmp    eax, [ebx]
10571 0000375D 746E                     je     short dscl_25
10572                                 dscl_23:
10573 0000375F 8815[99460000]           mov    [ds_drv], dl
10574                                 dscl_26:
10575 00003765 8B0B                     mov    ecx, [ebx]
10576 00003767 890D[2C480000]           mov    [prev_sec], ecx
10577 0000376D 8903                     mov    [ebx], eax
10578 0000376F E8E7030000               call   read_disk_sector
10579 00003774 733C                     jnc    short dscl_24
10580                                 dscl_rd_err:
10581                                   ;
10582                                   ;; Temporary, 15/12/2014
10583 00003776 88E0                     mov    al, ah ; error code
10584 00003778 66BF[0B48]               mov    di, err_code_str
10585 0000377C E8F6050000               call   write_hex
10586                                   ;
10587 00003781 BE[FC470000]             mov    esi, dskr_err ; drive not ready or read error
10588 00003786 E805030000               call   inputbox
10589 0000378B E845010000               call   getch
10590 00003790 E85C010000               call   restore_video_page
10591 00003795 29DB                     sub    ebx, ebx
10592 00003797 8A1D[99460000]           mov    bl, [ds_drv]
10593 0000379D C0E302                   shl    bl, 2
10594 000037A0 81C3[9B460000]           add    ebx, ds_sec
10595 000037A6 A1[2C480000]             mov    eax, [prev_sec]
10596 000037AB 8903                     mov    [ebx], eax
10597 000037AD E99FFDFFFF               jmp     dscl_getc
10598                                 dscl_24:
10599 000037B2 668B15[99460000]         mov    dx, [ds_drv]
10600 000037B9 31DB                     xor    ebx, ebx
10601 000037BB 88D3                     mov    bl, dl
```

```
10602 000037BD C0E302                   shl    bl, 2
10603 000037C0 81C3[9B460000]           add    ebx, ds_sec
10604 000037C6 8B03                     mov    eax, [ebx]
10605 000037C8 BE[D04A0000]             mov    esi, buffer
10606                                dscl_25:
10607 000037CD E834010000               call   display_sector
10608 000037D2 E805010000               call   save_video_page
10609 000037D7 E975FDFFFF                 jmp     dscl_getc
10610                                dscl_11:
10611 000037DC BE[D04A0000]             mov    esi, buffer
10612 000037E1 8A15[99460000]           mov    dl, [ds_drv]
10613 000037E7 28F6                     sub    dh, dh  ; 0 = first half of sector
10614 000037E9 E952FFFFFF                 jmp     dscl_ns
10615                                dscl_5:
10616 000037EE 663D003C                 cmp    ax, F2Key
10617 000037F2 750F                     jne    short dscl_6
10618 000037F4 E8E3000000               call   save_video_page
10619 000037F9 BE[EC470000]             mov    esi, F2_ib ; F2 (Change sector)
10620                                                       ; Inputbox address
10621                                   ;mov   byte [dscmd], 1
10622 000037FE E988FDFFFF                 jmp     dscl_ib
10623                                dscl_6:
10624 00003803 3C20                     cmp    al, SPACEKey
10625 00003805 0F84FAFEFFFF               je      dscl_oh
10626 0000380B 3C0D                     cmp    al, ENTERKey
10627 0000380D 0F84F2FEFFFF               je      dscl_oh
10628                                   ;
10629 00003813 663DE047                 cmp    ax, HOMEKey
10630 00003817 7504                     jne    short dscl_7
10631 00003819 31C0                     xor    eax, eax
10632 0000381B EBBF                     jmp    short dscl_11
10633                                dscl_7:
10634 0000381D 663DE04F                 cmp    ax, ENDKey
10635 00003821 7516                     jne    short dscl_8
10636 00003823 31DB                     xor    ebx, ebx
10637 00003825 8A1D[99460000]           mov    bl, [ds_drv]
10638 0000382B C0E302                   shl    bl, 2
10639 0000382E 81C3[EA480000]           add    ebx, disk_size
10640 00003834 8B03                     mov    eax, [ebx]
10641 00003836 48                       dec    eax
10642 00003837 EBA3                     jmp    short dscl_11
10643                                dscl_8:
10644 00003839 663DE051                 cmp    ax, PgDnKey
10645 0000383D 7541                     jne    short dscl_10
10646 0000383F E810000000               call   dscl_9
10647 00003844 40                       inc    eax
10648 00003845 39C8                     cmp    eax, ecx ; last sector
10649 00003847 0F86F3FEFFFF             jna    dscl_ns
10650 0000384D 31C0                     xor    eax, eax
```

```
10651 0000384F E911FFFFFF               jmp   dscl_26
10652                                 dscl_9:
10653 00003854 8A15[99460000]           mov   dl, [ds_drv]
10654 0000385A 30F6                     xor   dh, dh
10655 0000385C 29DB                     sub   ebx, ebx
10656 0000385E 88D3                     mov   bl, dl
10657 00003860 C0E302                   shl   bl, 2  ; *4
10658 00003863 81C3[EA480000]           add   ebx, disk_size
10659 00003869 8B0B                     mov   ecx, [ebx]
10660 0000386B 49                       dec   ecx
10661 0000386C 81EB[EA480000]           sub   ebx, disk_size
10662 00003872 81C3[9B460000]           add   ebx, ds_sec ; current sector
10663 00003878 8B03                     mov   eax, [ebx]
10664 0000387A BE[D04A0000]             mov   esi, buffer
10665 0000387F C3                         retn
10666                                 dscl_10:
10667 00003880 663DE049                 cmp   ax, PgUpKey
10668 00003884 0F85C7FCFFFF             jne   dscl_getc
10669 0000388A E8C5FFFFFF               call  dscl_9
10670 0000388F 48                       dec   eax
10671 00003890 39C8                     cmp   eax, ecx ; last sector
10672 00003892 0F86A8FEFFFF             jna   dscl_ns
10673 00003898 89C8                     mov   eax, ecx
10674 0000389A E9C6FEFFFF               jmp   dscl_26
10675
10676                                 dscl_exit:
10677                                   ; 12/02/2015 (unix386.s)
10678                                   ; 11/12/2014 (dsectrm2.s)
10679                                   ;
10680                                   ; Restore video page (before displaying sector)
10681 0000389F BF00800B00               mov   edi, 0B8000h
10682 000038A4 89FE                     mov   esi, edi
10683 000038A6 81C6C05D0000             add   esi, 5DC0h  ; 4000*6 (video page 6)
10684 000038AC B9E8030000               mov   ecx, 1000
10685 000038B1 F3A5                     rep   movsd
10686                                   ; Restore cursor position
10687 000038B3 668B15[20480000]         mov   dx, [cursor_posb] ; cursor position backup
10688 000038BA 668915[3C490000]         mov   [cursor_posn], dx ; for video page 0.
10689                                   ;
10690                                   ; Set cursor position
10691 000038C1 30DB                     xor   bl, bl  ; Video page 0
10692 000038C3 E8A2E1FFFF               call  set_cpos
10693                                   ; Show standard blinking text cursor
10694 000038C8 668B0D[22480000]         mov   cx, [cursor_shp]
10695 000038CF E8E1E1FFFF                 call set_ctype
10696                                 ;terminate:
10697                                   ;
10698                                   ;jmp   cpu_reset
10699                                   ;
```

```
10700 000038D4 C3                       retn
10701                                 ;hltsys:
10702                                 ;     hlt
10703                                 ;     jmp   short hltsys
10704
10705                                 getch:
10706                                       ; 22/02/2015 - dsectpm
10707 000038D5 B410                         mov   ah, 10h ; enhanced keyboard, read function
10708 000038D7 E9B2D6FFFF                   jmp   getc   ; jump to getchar
10709                                                     ; (getc_int, modified interrupt) routine
10710                                 ;getch:
10711                                       ; 18/02/2015 (unix386.s)
10712                                       ; This routine will be replaced with Retro UNIX 386
10713                                       ; version of Retro UNIX 8086 getch (tty input)
10714                                       ; routine, later... (multi tasking ability)
10715                                       ;push  esi
10716                                       ;push  ebx
10717                                       ;xor   ebx, ebx
10718                                       ;mov   bl, [ptty]  ; active_page
10719                                       ;mov   esi, ebx
10720                                       ;shl   si, 1
10721                                       ;add   esi, ttychr
10722                                 ;getch_1:
10723                                       ;mov   ax, word [esi]
10724                                 ;     mov   ax, [ttychr] ; video page 0 (tty0)
10725                                 ;     and   ax, ax
10726                                 ;     jz    short getch_2
10727                                 ;     mov   word [ttychr], 0
10728                                       ;mov   word [esi], 0
10729                                       ;pop   ebx
10730                                       ;pop   esi
10731                                 ;     retn
10732                                 ;getch_2:
10733                                       ;sti
10734                                 ;     hlt   ; not proper for multi tasking!
10735                                             ; (temporary halt for now)
10736                                             ; 'sleep' on tty
10737                                             ; will (must) be located here
10738                                 ;     nop
10739                                 ;     jmp   short getch_1
10740
10741                                 save_video_page:
10742                                       ; 12/02/2015 (unix386.s)
10743                                       ; 26/11/2014 (dsectrm2.s)
10744                                       ;
10745                                       ; Save video page
10746 000038DC BE00800B00                   mov   esi, 0B8000h
10747 000038E1 89F7                         mov   edi, esi
10748 000038E3 81C7606D0000                 add   edi, 6D60h  ; 4000*7 (video page 7)
```

```
10749 000038E9 B9E8030000                  mov    ecx, 1000
10750 000038EE F3A5                        rep    movsd
10751 000038F0 C3                          retn
10752
10753                                  restore_video_page:
10754                                      ; 12/02/2015 (unix386.s)
10755                                      ; 26/11/2014 (dsectrm2.s)
10756 000038F1 BE00800B00                  mov    esi, 0B8000h
10757 000038F6 89F7                        mov    edi, esi
10758 000038F8 81C6606D0000                add    esi, 6D60h  ; 4000*7 (video page 7)
10759 000038FE B9E8030000                  mov    ecx, 1000
10760 00003903 F3A5                        rep    movsd
10761 00003905 C3                          retn
10762
10763                                  display_sector:
10764                                      ; 19/02/2015
10765                                      ; 12/02/2015 (unix386.s)
10766                                      ; 6/11/2014 - 04/12/2014 (dsectrm2.s)
10767                                      ;
10768                                      ; display disk sector data (on tty0)
10769                                      ;
10770                                      ; INPUT ->
10771                                      ;     ESI = sector buffer offset
10772                                      ;           (sector size: 512 bytes)
10773                                      ;     EAX = sector number
10774                                      ;     DL = drive number (0,1,2,3,4,5,6)
10775                                      ;     DH = portion control byte
10776                                      ;             (0= first half of the sector,
10777                                      ;             >0= second half of the sector)
10778                                      ; OUTPUT ->
10779                                      ;     Video page 0 (0B8000h) will be filled
10780                                      ;     with sector data
10781                                      ;     (ESI points to byte 256 of the buffer
10782                                      ;     or end of the buffer)
10783                                      ;
10784                                      ; Modified registers: eax, edx, ecx, ebx, esi, edi
10785                                      ;
10786                                      ;
10787                                      ;xor   ecx, ecx ; reset for cx loop counts
10788 00003906 C605[98460000]01            mov    byte [inds], 1 ; for ENTER key handling
10789                                      ;
10790 0000390D 50                          push   eax
10791 0000390E E81D010000                  call   clear_frame
10792 00003913 58                          pop    eax
10793                                  dsfh:
10794 00003914 31DB                        xor    ebx, ebx
10795 00003916 08F6                        or     dh, dh
10796 00003918 7402                        jz     short dsfh1
10797 0000391A B310                        mov    bl, 10h
```

```
10798                                  dsfh1:
10799 0000391C 881D[B3460000]                mov   [paragr], bl ; Paragraph (16 bytes)
10800                                        ;
10801 00003922 88D3                          mov   bl, dl
10802 00003924 C0E302                        shl   bl, 2  ; *4
10803 00003927 81C3[B4460000]                add   ebx, drv_names
10804 0000392D 8B13                          mov   edx, [ebx]
10805 0000392F 8915[D5460000]                mov   [drv_name], edx
10806 00003935 E8BD000000                    call  dwordtohex
10807 0000393A 8915[E3460000]                mov   [sector_num], edx
10808 00003940 A3[E7460000]                  mov   [sector_num+4], eax
10809 00003945 B001                          mov   al, 1
10810 00003947 B43F                          mov   ah, 3Fh ; cyan background, white forecolor
10811 00003949 BB[CC460000]                  mov   ebx, dpheader
10812 0000394E E8CA000000                    call  print_line
10813 00003953 B015                          mov   al, 21
10814                                        ;mov  ah, 3Fh ; cyan background, white forecolor
10815 00003955 BB[43470000]                  mov   ebx, dpfooter1
10816 0000395A E8BE000000                    call  print_line
10817 0000395F B016                          mov   al, 22
10818                                        ;mov  ah, 3Fh ; cyan background, white forecolor
10819 00003961 BB[8D470000]                  mov   ebx, dpfooter2
10820 00003966 E8B2000000                    call  print_line
10821                                  ds1:
10822 0000396B B910000000                    mov   ecx, 16
10823                                  ds2:
10824 00003970 A0[B3460000]                  mov   al, [paragr]
10825 00003975 E83D000000                    call  bytetohex
10826 0000397A 66A3[F9460000]                mov   [sdline_1], ax
10827                                        ;
10828 00003980 51                            push  ecx
10829 00003981 B110                          mov   cl, 16
10830 00003983 BF[01470000]                  mov   edi, sdline_2
10831                                  ds3:
10832 00003988 AC                            lodsb
10833 00003989 E829000000                    call  bytetohex
10834 0000398E 66AB                          stosw
10835 00003990 47                            inc   edi
10836 00003991 E2F5                          loop  ds3
10837 00003993 83EE10                        sub   esi, 16
10838 00003996 47                            inc   edi
10839 00003997 B110                          mov   cl, 16
10840 00003999 F3A4                          rep   movsb
10841 0000399B 59                            pop   ecx
10842 0000399C B013                          mov   al, 19 ; line (row) 3 to 24
10843 0000399E 28C8                          sub   al, cl
10844 000039A0 B407                          mov   ah, 07h ; Black background, light gray forecolor
10845 000039A2 BB[F3460000]                  mov   ebx, sdline
10846 000039A7 E85D000000                    call  print_line_80 ; 04/12/2014
```

```
10847 000039AC E201                     loop   ds4
10848 000039AE C3                       retn
10849                                 ds4:
10850 000039AF FE05[B3460000]           inc    byte [paragr]
10851 000039B5 EBB9                     jmp    short ds2
10852
10853                                 ; Convert binary number to hexadecimal string
10854                                 ; 01/12/2014
10855                                 ; 25/11/2014
10856                                 ;
10857                                 bytetohex:
10858                                       ; INPUT ->
10859                                       ;      AL = byte (binary number)
10860                                       ; OUTPUT ->
10861                                       ;      AX = hexadecimal string
10862                                       ;
10863 000039B7 53                           push   ebx
10864 000039B8 31DB                         xor    ebx, ebx
10865 000039BA 88C3                         mov    bl, al
10866 000039BC C0EB04                       shr    bl, 4
10867 000039BF 8A9B[88460000]               mov    bl, [ebx+hexchrs]
10868 000039C5 86D8                         xchg   bl, al
10869 000039C7 80E30F                       and    bl, 0Fh
10870 000039CA 8AA3[88460000]               mov    ah, [ebx+hexchrs]
10871 000039D0 5B                           pop    ebx
10872 000039D1 C3                           retn
10873
10874                                 wordtohex:
10875                                       ; INPUT ->
10876                                       ;      AX = word (binary number)
10877                                       ; OUTPUT ->
10878                                       ;      EAX = hexadecimal string
10879                                       ;
10880 000039D2 53                           push   ebx
10881 000039D3 31DB                         xor    ebx, ebx
10882 000039D5 86E0                         xchg   ah, al
10883 000039D7 6650                         push   ax
10884 000039D9 88E3                         mov    bl, ah
10885 000039DB C0EB04                       shr    bl, 4
10886 000039DE 8A83[88460000]               mov    al, [ebx+hexchrs]
10887 000039E4 88E3                         mov    bl, ah
10888 000039E6 80E30F                       and    bl, 0Fh
10889 000039E9 8AA3[88460000]               mov    ah, [ebx+hexchrs]
10890 000039EF C1E010                       shl    eax, 16
10891 000039F2 6658                         pop    ax
10892 000039F4 5B                           pop    ebx
10893 000039F5 EBC0                         jmp    short bytetohex
10894                                       ;mov   bl, al
10895                                       ;shr   bl, 4
```

```
10896                                         ;mov   bl, [ebx+hexchrs]
10897                                         ;xchg  bl, al
10898                                         ;and   bl, 0Fh
10899                                         ;mov   ah, [ebx+hexchrs]
10900                                         ;pop   ebx
10901                                         ;retn
10902
10903                                 dwordtohex:
10904                                         ; INPUT ->
10905                                         ;      EAX = dword (binary number)
10906                                         ; OUTPUT ->
10907                                         ;      EDX:EAX = hexadecimal string
10908                                         ;
10909 000039F7 50                             push   eax
10910 000039F8 C1E810                         shr    eax, 16
10911 000039FB E8D2FFFFFF                     call   wordtohex
10912 00003A00 89C2                           mov    edx, eax
10913 00003A02 58                             pop    eax
10914 00003A03 E8CAFFFFFF                     call   wordtohex
10915 00003A08 C3                             retn
10916
10917                                 print_line_80:
10918                                         ; 04/12/2014
10919                                         ; al = line (0 to 24)
10920                                         ; ah = color attributes
10921                                         ; ebx = 80 chars string address
10922 00003A09 E86E000000                     call   get_lpos
10923 00003A0E 51                             push   ecx
10924 00003A0F B950000000                     mov    ecx, 80
10925                                 pl80:
10926 00003A14 8A03                           mov    al, [ebx]
10927 00003A16 43                             inc    ebx
10928 00003A17 66AB                           stosw
10929 00003A19 E2F9                           loop   pl80
10930 00003A1B 59                             pop    ecx
10931 00003A1C C3                             retn
10932
10933                                 print_line:
10934                                         ; al = line (0 to 24)
10935                                         ; ah = color attributes
10936                                         ; ebx = ASCIIZ string address
10937 00003A1D E85A000000                     call   get_lpos
10938 00003A22 56                             push   esi
10939 00003A23 89DE                           mov    esi,ebx
10940                                 prl1:
10941 00003A25 AC                             lodsb
10942 00003A26 20C0                           and    al, al
10943 00003A28 7404                           jz     short prl2
10944 00003A2A 66AB                           stosw
```

```
10945 00003A2C EBF7                     jmp    short prl1
10946                                 prl2:
10947 00003A2E 5E                       pop    esi
10948 00003A2F C3                       retn
10949
10950                                 clear_frame:
10951 00003A30 30C0                     xor    al, al ; Line 0
10952 00003A32 E833000000               call   clear_line
10953 00003A37 B001                     mov    al, 1
10954 00003A39 B43F                     mov    ah, 3Fh ; cyan background, white forecolor
10955 00003A3B E82C000000               call   fill_color
10956 00003A40 B001                     mov    al, 1
10957                                 dscf0:
10958 00003A42 6640                     inc    ax
10959 00003A44 6650                     push   ax
10960 00003A46 E81F000000               call   clear_line
10961 00003A4B 6658                     pop    ax
10962 00003A4D 3C13                     cmp    al, 19
10963 00003A4F 72F1                     jb     short dscf0
10964                                   ;inc   al ; line 20
10965 00003A51 B43F                     mov    ah, 3Fh
10966                                 dscf1:
10967 00003A53 FEC0                     inc    al
10968 00003A55 6650                     push   ax
10969 00003A57 E810000000               call   fill_color
10970 00003A5C 6658                     pop    ax
10971 00003A5E 3C17                     cmp    al, 23
10972 00003A60 72F1                     jb     short dscf1
10973 00003A62 FEC0                     inc    al
10974 00003A64 E801000000               call   clear_line
10975 00003A69 C3                       retn
10976
10977                                 clear_line:
10978 00003A6A 30E4                     xor    ah, ah ; blank
10979                                 fill_color:
10980                                   ; al = line (0 to 24)
10981                                   ; ah = color attributes
10982 00003A6C E80B000000               call   get_lpos
10983 00003A71 B950000000               mov    ecx, 80 ; 23/02/2015
10984 00003A76 B020                     mov    al, 20h ; space/blank
10985 00003A78 F366AB                   rep    stosw
10986 00003A7B C3                       retn
10987
10988                                 get_lpos:  ; Get line position in video buffer
10989 00003A7C 6650                     push   ax
10990 00003A7E 31FF                     xor    edi, edi
10991 00003A80 B4A0                     mov    ah, 80*2
10992 00003A82 F6E4                     mul    ah
10993 00003A84 6689C7                   mov    di, ax
```

```
10994 00003A87 6658                     pop    ax
10995 00003A89 81C700800B00             add    edi, 0B8000h ; video page 0
10996 00003A8F C3                       retn
10997
10998                                 inputbox:
10999                                   ; 12/02/2015 (unix386.s)
11000                                   ; 27/11/2014 (dsectrm2.s)
11001                                   ; Show an input box for user/keyboard input
11002                                   ; INPUT ->
11003                                   ;      ESI = input structure address
11004                                   ; OUTPUT ->
11005                                   ;      DX =  cursor position for input
11006                                   ;      input box will be displayed (on tty0)
11007                                   ;
11008                                   ; Modified registers: ax, ebx, ecx, dx, esi, edi
11009                                   ;
11010
11011 00003A90 C605[98460000]00         mov    byte [inds], 0 ; for ENTER key handling
11012 00003A97 31C9                     xor    ecx, ecx
11013                                   ;xor   cx, cx   ; ecx = 0
11014 00003A99 BB00800B00               mov    ebx, 0B8000h
11015 00003A9E 66B81850                 mov    ax, 5018h ; 80, 24
11016 00003AA2 668B16                   mov    dx, [esi] ; box width (dl)
11017                                                    ; box height (dh)
11018 00003AA5 28F0                     sub    al, dh
11019 00003AA7 D0E8                     shr    al, 1
11020 00003AA9 A2[DD470000]             mov    [ibcp+1], al ; row
11021 00003AAE F6E4                     mul    ah
11022 00003AB0 66D1E0                   shl    ax, 1  ; char + attribute
11023                                   ;mov   bx, ax
11024 00003AB3 6601C3                   add    bx, ax
11025 00003AB6 B050                     mov    al, 80
11026 00003AB8 28D0                     sub    al, dl
11027 00003ABA D0E8                     shr    al, 1
11028 00003ABC A2[DC470000]             mov    [ibcp], al ; column
11029 00003AC1 D0E0                     shl    al, 1  ; char + attribute
11030 00003AC3 28E4                     sub    ah, ah
11031 00003AC5 6601C3                   add    bx, ax
11032 00003AC8 8A6605                   mov    ah, [esi+5] ; color attributes
11033 00003ACB B020                     mov    al, 20h      ; space/blank
11034 00003ACD 88F1                     mov    cl, dh ; height
11035                                 ib0:
11036 00003ACF 6651                     push   cx
11037 00003AD1 88D1                     mov    cl, dl
11038 00003AD3 89DF                     mov    edi, ebx
11039 00003AD5 F366AB                   rep    stosw
11040 00003AD8 6659                     pop    cx
11041 00003ADA 6681C3A000               add    bx, 80*2 ; number of columns * 2
11042 00003ADF E2EE                     loop   ib0
```

```
11043                                          ;
11044 00003AE1 BF00800B00                      mov   edi, 0B8000h
11045 00003AE6 A0[DD470000]                    mov   al, [ibcp+1] ; row position
11046 00003AEB 024602                          add   al, [esi+2] ; label offset (row)
11047 00003AEE A2[DD470000]                    mov   [ibcp+1], al
11048 00003AF3 B4A0                            mov   ah, 80*2
11049 00003AF5 F6E4                            mul   ah
11050 00003AF7 6601C7                          add   di, ax
11051 00003AFA A0[DC470000]                    mov   al, [ibcp] ; column position
11052 00003AFF 024603                          add   al, [esi+3] ; label offset (column)
11053 00003B02 A2[DC470000]                    mov   [ibcp], al
11054 00003B07 30E4                            xor   ah, ah
11055 00003B09 D0E0                            shl   al, 1
11056 00003B0B 6601C7                          add   di, ax
11057 00003B0E 89F3                            mov   ebx, esi
11058 00003B10 83C606                          add   esi, 6 ; Label offset
11059                                  ib2:
11060 00003B13 AC                              lodsb
11061 00003B14 08C0                            or    al, al
11062 00003B16 7406                            jz    short ib3
11063 00003B18 AA                              stosb
11064 00003B19 47                              inc   edi
11065 00003B1A FEC1                            inc   cl
11066 00003B1C EBF5                            jmp   short ib2
11067                                  ib3:
11068 00003B1E 000D[DC470000]                  add   [ibcp], cl ; column position
11069 00003B24 893D[ED460000]                  mov   [current_txtpos], edi
11070                                          ;
11071 00003B2A 8A4B04                          mov   cl, [ebx+4] ; input char count
11072 00003B2D 08C9                            or    cl, cl
11073 00003B2F 740E                            jz    short ib5 ; message box (no input)
11074 00003B31 B020                            mov   al, 20h
11075 00003B33 B407                            mov   ah, 07h ; black background
11076                                                        ; light gray fore color
11077                                  ib4:
11078 00003B35 F366AB                          rep   stosw
11079                                          ;
11080 00003B38 668B15[DC470000]                mov   dx, [ibcp] ; cursor position
11081                                  ib5:
11082 00003B3F C3                              retn
11083
11084                                  hide_cursor:
11085                                          ; 12/02/2015 (unix386.s)
11086                                            ; 01/12/2014 (dsectrm2.s)
11087                                          ;CH = cursor start line (bits 0-4)
11088                                          ;     and options (bits 5-7).
11089                                          ;CL = bottom cursor line (bits 0-4).
11090                                          ; when bit 5 of CH is set to 0, the cursor is visible.
11091                                          ; when bit 5 is 1, the cursor is not visible.
```

```
11092                                       ; hide blinking text cursor:
11093 00003B40 B520                           mov  ch, 32
11094 00003B42 EB10                         jmp    short hc_sc
11095
11096                                   show_cursor:
11097                                       ; 12/02/2015 (unix386.s)
11098                                         ; 01/12/2014 - 13/12/2014 (dsectrm2.s)
11099                                       ; dh = row
11100                                       ; dl = column
11101 00003B44 668915[3C490000]             mov    [cursor_posn], dx
11102 00003B4B 28DB                         sub    bl, bl ; video page 0
11103 00003B4D E818DFFFFF                   call   set_cpos
11104                                       ;
11105                                       ;show blinking text cursor
11106 00003B52 B50D                         mov    ch, 13 ; 16/02/2015
11107                                   hc_sc:
11108 00003B54 B10F                           mov  cl, 15 ; 16/02/2015
11109 00003B56 E95ADFFFFF                         jmp    set_ctype
11110
11111                                   read_disk_sector:
11112                                       ; 21/02/2015
11113                                       ; 16/02/2015 (unix386.s)
11114                                       ; 01/12/2014 - 18/01/2015 (dsectrm2.s)
11115 00003B5B 31DB                         xor    ebx, ebx
11116 00003B5D 8A1D[99460000]               mov    bl, [ds_drv]
11117 00003B63 89DE                         mov    esi, ebx ; 26/02/2015
11118 00003B65 88DA                         mov    dl, bl
11119 00003B67 80FA02                       cmp    dl, 2
11120 00003B6A 7203                         jb     short rd0
11121 00003B6C 80C27E                       add    dl, 7Eh  ; 80h, 81h, 82h, 83h
11122                                   rd0:
11123 00003B6F 8815[13490000]               mov    [drv], dl
11124 00003B75 81C3[24480000]               add    ebx, drv_status
11125 00003B7B 8A33                         mov    dh, [ebx]
11126                                   rd1:
11127 00003B7D 80FEF0                       cmp    dh, 0F0h ; 18/1/2015
11128 00003B80 F5                           cmc
11129 00003B81 0F82CF000000                   jc     rd_fails
11130                                       ;
11131                                       ;xor   ebx, ebx
11132                                       ;mov   bl, [ds_drv]
11133                                       ;;mov  esi, ebx
11134 00003B87 89F3                         mov    ebx, esi ; 26/02/2015
11135 00003B89 C0E302                       shl    bl, 2
11136 00003B8C 81C3[9B460000]               add    ebx, ds_sec
11137 00003B92 8B03                         mov    eax, [ebx]
11138 00003B94 81EB[9B460000]               sub    ebx, ds_sec
11139 00003B9A 81C3[EA480000]               add    ebx, disk_size
11140 00003BA0 3B03                         cmp    eax, [ebx] ; Last sector + 1 (number of secs.)
```

```
11141 00003BA2 F5                       cmc
11142 00003BA3 7247                       jc      short rd_lba_fails
11143                                   ; 02/02/2015
11144 00003BA5 F6C601                   test   dh, 1 ; LBA ready ?
11145 00003BA8 7443                       jz      short rd_chs
11146                                 rd_lba:
11147                                   ; LBA read (with temporary function)
11148                                   ;((Retro UNIX 386 v1 - DISK I/O Test))
11149 00003BAA 81C6[24480000]           add    esi, drv_status
11150 00003BB0 80268F                   and    byte [esi], 8Fh ; clear error bits
11151                                   ;
11152                                   ;mov   ebx, eax
11153                                   ;mov   cx, bx
11154                                   ;shr   ebx, 16
11155                                   ;;mov  di, cs
11156                                   ;;mov  es, di
11157                                   ;mov   edi, buffer
11158                                   ; 21/02/2015 (unix386.s, 32 bit modification)
11159 00003BB3 89C1                     mov    ecx, eax
11160 00003BB5 BB[D04A0000]             mov    ebx, buffer
11161 00003BBA 8A15[13490000]           mov    dl, [drv]
11162 00003BC0 C605[6C3C0000]04         mov    byte [retry_count], 4
11163                                 rd_lba_retry:
11164 00003BC7 B41B                     mov    ah, 1Bh ; Temporary/Private function
11165 00003BC9 B001                     mov    al, 1
11166                                   ;int   13h
11167 00003BCB E8D7F0FFFF               call   int13h
11168 00003BD0 731A                     jnc    short rd_lba_ok
11169 00003BD2 80FC80                   cmp    ah, 80h ; time out ?
11170 00003BD5 7414                     je     short rd_lba_rfails
11171 00003BD7 FE0D[6C3C0000]           dec    byte [retry_count]
11172 00003BDD 740C                     jz     short rd_lba_rfails
11173 00003BDF B40D                     mov    ah, 0Dh ; Alternate reset
11174                                   ;int   13h
11175 00003BE1 E8C1F0FFFF                 call int13h
11176 00003BE6 73DF                     jnc     short rd_lba_retry
11177 00003BE8 800EF0                   or     byte [esi], 0F0h ; drive not ready !
11178                                 rd_lba_rfails:
11179 00003BEB F9                       stc
11180                                 rd_lba_fails:
11181                                 rd_lba_ok:
11182 00003BEC C3                       retn
11183                                   ;
11184                                   ; CHS read (convert LBA address to CHS values)      ;
11185                                 rd_chs:
11186 00003BED D1E6                     shl    esi, 1
11187 00003BEF 89F3                     mov    ebx, esi
11188 00003BF1 31D2                     xor    edx, edx ; 0
11189 00003BF3 29C9                     sub    ecx, ecx
```

```
11190 00003BF5 81C3[DE480000]               add    ebx, spt
11191 00003BFB 668B0B                       mov    cx, [ebx] ; sector per track
11192                                       ; EDX:EAX = LBA
11193 00003BFE F7F1                         div    ecx
11194 00003C00 88D1                         mov    cl, dl ; sector number - 1
11195 00003C02 FEC1                         inc    cl     ; sector number (1 based)
11196 00003C04 6651                         push   cx
11197 00003C06 89F3                         mov    ebx, esi
11198 00003C08 81C3[D2480000]               add    ebx, heads
11199 00003C0E 668B0B                       mov    cx, [ebx] ; heads
11200 00003C11 31D2                         xor    edx, edx
11201                                       ; EAX = cylinders * heads + head
11202 00003C13 F7F1                         div    ecx
11203 00003C15 6659                         pop    cx     ; sector number
11204 00003C17 88D6                         mov    dh, dl ; head number
11205 00003C19 8A15[13490000]               mov    dl, [drv]
11206 00003C1F 88C5                         mov    ch, al ; cylinder (bits 0-7)
11207 00003C21 C0E406                       shl    ah, 6
11208 00003C24 08E1                         or     cl, ah ; cylinder (bits 8-9)
11209                                              ; sector (bits 0-7)
11210                                       ;;push cs
11211                                       ;;pop  es
11212 00003C26 BB[D04A0000]                 mov    ebx, buffer
11213                                              ; CL = sector (bits 0-6)
11214                                              ;      cylinder (bits 7-8 -> bits 8-9)
11215                                              ; CH = cylinder (bits 0-7)
11216                                              ; DH = head
11217                                              ; DL = drive
11218                                       ; 02/02/2015
11219 00003C2B D1EE                         shr    esi, 1 ; drive index (byte alignment)
11220 00003C2D 81C6[24480000]               add    esi, drv_status
11221 00003C33 80268F                       and    byte [esi], 8Fh ; clear error bits
11222                                       ;
11223 00003C36 C605[6C3C0000]04             mov    byte [retry_count], 4
11224                                 rd_retry:
11225 00003C3D B402                         mov    ah, 02h ; read sectors
11226 00003C3F B001                         mov    al, 1 ; sector count
11227                                       ;int   13h
11228 00003C41 E861F0FFFF                   call   int13h
11229 00003C46 7323                         jnc    short rd_ok
11230 00003C48 80FC80                       cmp    ah, 80h ; time out ?
11231 00003C4B 7408                         je     short rd_rfails
11232 00003C4D FE0D[6C3C0000]               dec    byte [retry_count]
11233 00003C53 7502                         jnz    short rd_reset
11234                                 rd_rfails:
11235 00003C55 F9                           stc
11236                                 rd_fails:
11237 00003C56 C3                           retn
11238                                 rd_reset:
```

```
11239                                          ; 02/02/2015
11240 00003C57 28E4                            sub    ah, ah
11241 00003C59 80FA80                          cmp    dl, 80h
11242 00003C5C 7202                            jb     rd_fd_reset
11243 00003C5E B40D                            mov    ah, 0Dh ; Alternate reset
11244                                  rd_fd_reset:
11245                                          ;int   13h
11246 00003C60 E842F0FFFF                        call int13h
11247 00003C65 73D6                            jnc     short rd_retry
11248 00003C67 800EF0                          or     byte [esi], 0F0h ; drive not ready !
11249 00003C6A F9                              stc
11250                                  rd_ok:
11251 00003C6B C3                              retn
11252
11253                                  retry_count: ; 16/12/2014
11254 00003C6C 00                              db     0
11255                                  ;rd_lba:
11256                                  ;     mov    [lba], eax
11257                                  ;     mov    ax, cs
11258                                  ;     mov    [tbuff+2], ax
11259                                  ;     mov    esi, disk_pkt
11260                                  ;     ;mov   dl, [drv]
11261                                  ;     mov    ah, 42h ; Extended read
11262                                  ;     int    13h
11263                                  ;     retn
11264
11265                                  ; 16/02/2015 (unix386.s)
11266                                  ; 19/12/2014 (dsectrm2.s)
11267                                  clear_screen:
11268                                          ;;mov ah, 0Fh ; get video mode
11269                                            ;;int 10h
11270                                          ;;; al = video mode
11271                                            ;;mov ah, 0   ; set video mode (clears screen)
11272                                            ;;int 10h
11273                                            ;;retn
11274                                          ; 19/12/2014
11275                                          ;push  es
11276 00003C6D 57                              push   edi
11277                                          ;mov   di, 0B800h
11278                                          ;mov   es, di
11279                                          ;sub   di, di
11280 00003C6E BF00800B00                      mov    edi, 0B8000h
11281 00003C73 B9E8030000                      mov    ecx, 1000
11282                                          ;mov cx, 2000
11283                                          ;mov   ax, 0720h ; light gray char space (blank)
11284 00003C78 B820072007                      mov    eax,07200720h
11285                                          ;rep   stosw
11286 00003C7D F3AB                            rep    stosd
11287 00003C7F 5F                              pop    edi
```

```
11288                                   ;pop   es
11289 00003C80 6631D2                   xor    dx, dx   ; column 0, row 0
11290 00003C83 28DB                     sub    bl, bl ; 17/02/2015 - video page 0
11291 00003C85 E9E0DDFFFF               jmp    set_cpos ; set cursor position
11292
11293                                 rfdp_err:
11294                                   ;23/12/2014
11295 00003C8A C3                       retn
11296
11297                                 dskprm:
11298                                   ; 20/02/2015
11299                                   ; 16/02/2015 (unix386.s)
11300                                   ; 19/12/2014 - 23/12/2014 (dsectrm2.s)
11301                                   ; DISPLAY DISK PARAMETERS TABLE
11302                                   ;
11303                                   ; INPUT -> DL = Disk/Drive #
11304                                   ;
11305 00003C8B 8815[13490000]           mov    byte [drv], dl  ; 0,1,80h,81h,82h,83h
11306                                   ;
11307 00003C91 B408                     mov    ah, 08h      ; Return drive parameters (conventional)
11308                                     ;int       13h
11309 00003C93 E80FF0FFFF               call   int13h
11310 00003C98 72F0                     jc     short rfdp_err
11311                                   ;
11312 00003C9A 6653                     push   bx ; 20/02/2015
11313                                   ; 23/12/2014
11314 00003C9C E8CCFFFFFF               call   clear_screen    ; clear video page 0
11315 00003CA1 665B                     pop    bx ; 20/02/2015
11316                                   ;
11317 00003CA3 F605[13490000]80           test byte [drv], 80h
11318 00003CAA 0F85E9000000               jnz     print_hdpt
11319 00003CB0 80C330                   add    bl, 30h      ; '0'
11320 00003CB3 881D[E83F0000]             mov     byte [flpdtype], bl
11321                                                          ; floppy disk drive type
11322                                                       ; (1=360K, 2=1.2M, 3=720K, 4=1.44M)
11323                                 print_flpdpt:
11324                                   ; 16/02/2015 (unix386.s)
11325                                   ; Writing the Diskette Parameter Table on screen
11326                                   ;mov   si, di
11327 00003CB9 89FE                     mov    esi, edi
11328                                   ;mov   ax, es
11329                                   ;mov   ds, ax
11330                                   ;mov   ax, cs
11331                                   ;mov   es, ax
11332 00003CBB AC                       lodsb  ; bits 0-3: SRT step rate time
11333                                          ; bits 4-7: head unload time
11334 00003CBC BF[32400000]             mov    edi, rSrtHdUnld
11335 00003CC1 E8B1000000               call   write_hex
11336 00003CC6 AC                       lodsb  ; bit 0: 1=use DMA
```

```
11337                                           ; bits 2-7: head load time
11338 00003CC7 BF[7A400000]                     mov   edi, rDmaHdLd
11339 00003CCC E8A6000000                       call  write_hex
11340 00003CD1 AC                               lodsb ; 55-ms increments
11341                                           ; before turning disk motor off
11342 00003CD2 BF[C1400000]                     mov   edi, bMotorOff
11343 00003CD7 E89B000000                       call  write_hex
11344 00003CDC AC                               lodsb ; sector size
11345                                           ; (0=128, 1=256, 2=512, 3=1024)
11346 00003CDD BF[06410000]                     mov   edi, bSectSize
11347 00003CE2 E890000000                       call  write_hex
11348 00003CE7 AC                               lodsb ; EOT (last sector on a track)
11349 00003CE8 BF[32410000]                     mov   edi, bLastTrack
11350 00003CED E885000000                       call  write_hex
11351 00003CF2 AC                               lodsb ; gap length
11352                                           ; for read/write operations
11353 00003CF3 BF[4E410000]                     mov   edi, bGapLen
11354 00003CF8 E87A000000                       call  write_hex
11355 00003CFD AC                               lodsb ; DTL (Data Transfer Length)
11356                                           ; max transfer when length not set
11357 00003CFE BF[6A410000]                     mov   edi, bDTL
11358 00003D03 E86F000000                       call  write_hex
11359 00003D08 AC                               lodsb ; gap length for format operation
11360 00003D09 BF[86410000]                     mov   edi, bGapFmt
11361 00003D0E E864000000                       call  write_hex
11362 00003D13 AC                               lodsb ; fill character for format
11363                                           ; (normally F6H)
11364 00003D14 BF[A2410000]                     mov   edi, bFillChar
11365 00003D19 E859000000                       call  write_hex
11366 00003D1E AC                               lodsb ; head-settle time
11367                                           ; (in milliseconds)
11368 00003D1F BF[CD410000]                     mov   edi, bHdSettle
11369 00003D24 E84E000000                       call  write_hex
11370 00003D29 AC                               lodsb ; motor-startup time
11371                                           ; (in 1/8th-second intervals)
11372 00003D2A BF[F6410000]                     mov   edi, bMotorOn
11373 00003D2F E843000000                       call  write_hex
11374                                           ; 19/12/2014
11375                                           ; (extension, not in original bios function)
11376 00003D34 AC                               lodsb ; Max. track number
11377 00003D35 BF[30420000]                     mov   edi, bMaxTrack
11378 00003D3A E838000000                       call  write_hex
11379 00003D3F AC                               lodsb ; Data transfer rate
11380 00003D40 BF[4C420000]                     mov   edi, bDataRate
11381 00003D45 E82D000000                       call  write_hex
11382                                           ;
11383                                           ;mov   ax, cs
11384                                           ;mov   ds, ax
11385                                            ;;mov     al, byte [ds_drv]
```

```
11386 00003D4A A0[13490000]                mov    al, [drv] ; 11/01/2015
11387 00003D4F 0430                        add    al, 30h ; '0'
11388 00003D51 A2[B13F0000]                  mov     byte [flpdnum], al
11389 00003D56 BE[A93F0000]                  mov     esi, FLPDPT
11390 00003D5B E831020000                  call   print_msg
11391 00003D60 C3                          retn
11392
11393                                  write_dhex:
11394                                        ; 16/02/2015 (unix386.s)
11395 00003D61 88E3                        mov    bl, ah
11396 00003D63 D0EB                          shr     bl, 1
11397 00003D65 D0EB                          shr     bl, 1
11398 00003D67 D0EB                          shr     bl, 1
11399 00003D69 D0EB                          shr     bl, 1
11400 00003D6B E818000000                    call    dhgd
11401 00003D70 88E3                        mov    bl, ah
11402 00003D72 E811000000                  call   dhgd
11403                                  write_hex:
11404 00003D77 88C3                        mov    bl, al
11405 00003D79 D0EB                          shr     bl, 1
11406 00003D7B D0EB                          shr     bl, 1
11407 00003D7D D0EB                          shr     bl, 1
11408 00003D7F D0EB                          shr     bl, 1
11409 00003D81 E802000000                  call   dhgd
11410 00003D86 88C3                        mov    bl, al
11411                                      ;call  dhgd
11412                                      ;retn
11413                                  dhgd:
11414 00003D88 6650                        push   ax
11415                                      ; 16/02/2015 (unix386.s, 32 bit modifications)
11416                                      ;and   bx, 0Fh
11417                                        ;add    bx, hex_digits
11418                                        ;mov    al, byte [CS:BX]
11419 00003D8A 83E30F                      and    ebx, 0Fh
11420 00003D8D 81C3[88460000]              add    ebx, hex_digits
11421 00003D93 8A03                        mov    al, [ebx]
11422 00003D95 AA                          stosb
11423 00003D96 6658                        pop    ax
11424 00003D98 C3                          retn
11425
11426                                  print_hdpt:
11427                                      ; 20/02/2015
11428                                      ; 16/02/2015 (unix386.s)
11429                                      ; 23/12/2014 - 11/01/2015 (dsectrm2.s)
11430                                      ;; ES:DI = DPT address
11431                                      ;
11432                                      ;;mov  si, HD0_DPT
11433 00003D99 BE00000900                  mov    esi, (DPT_SEGM*16)+HD0_DPT
11434 00003D9E 8A1D[13490000]              mov    bl, [drv] ; 20/02/2015
```

```
11435 00003DA4 83E303                   and    ebx, 3
11436 00003DA7 88D8                     mov    al, bl
11437 00003DA9 0402                     add    al, 2
11438 00003DAB A2[99460000]             mov    [ds_drv], al
11439                                   ;
11440 00003DB0 C0E305                   shl    bl, 5 ; * 32
11441 00003DB3 01DE                     add    esi, ebx
11442                                   ;;mov  ax, DPT_SEGM
11443                                   ;;mov  ds, ax
11444                                   ;;mov  ax, cs
11445                                   ;;mov  es, ax
11446 00003DB5 807E03A0                 cmp    byte [esi+3], 0A0h ; Translated table
11447 00003DB9 0F84EC000000               je      print_thdpt       ; indicator
11448                                   ;
11449                                   ; Writing Fixed Disk Parameter Table on screen
11450 00003DBF 66AD                     lodsw  ; Number of Cylinders
11451 00003DC1 BF[BF420000]             mov    edi, cylnum
11452 00003DC6 E896FFFFFF               call   write_dhex
11453 00003DCB AC                       lodsb  ; Number of Heads
11454 00003DCC BF[DC420000]             mov    edi, headnum
11455 00003DD1 E8A1FFFFFF               call   write_hex
11456 00003DD6 AC                       lodsb  ; Reserved
11457 00003DD7 BF[F7420000]             mov    edi, rsvd3
11458 00003DDC E896FFFFFF               call   write_hex
11459 00003DE1 AC                       lodsb  ; Reserved
11460 00003DE2 BF[12430000]             mov    edi, rsvd4
11461 00003DE7 E88BFFFFFF               call   write_hex
11462 00003DEC 66AD                     lodsw  ; Precompensation (Obsolete)
11463 00003DEE BF[2D430000]             mov    edi, pcompnum
11464 00003DF3 E869FFFFFF               call   write_dhex
11465 00003DF8 AC                       lodsb  ; Reserved
11466 00003DF9 BF[4A430000]             mov    edi, rsvd7
11467 00003DFE E874FFFFFF               call   write_hex
11468 00003E03 AC                       lodsb  ; Drive Control Byte
11469 00003E04 BF[65430000]             mov    edi, dcbnum
11470 00003E09 E869FFFFFF               call   write_hex
11471 00003E0E 66AD                     lodsw  ; Reserved
11472 00003E10 BF[80430000]             mov    edi, rsvd9
11473 00003E15 E847FFFFFF               call   write_dhex
11474 00003E1A AC                       lodsb  ; Reserved
11475 00003E1B BF[9D430000]             mov    edi, rsvd11
11476 00003E20 E852FFFFFF               call   write_hex
11477 00003E25 66AD                     lodsw  ; Landing Zone (Obsolete)
11478 00003E27 BF[B8430000]             mov    edi, lzonenum
11479 00003E2C E830FFFFFF               call   write_dhex
11480 00003E31 AC                       lodsb  ; Sectors per Track
11481 00003E32 BF[D5430000]             mov    edi, psptnum
11482 00003E37 E83BFFFFFF               call   write_hex
11483 00003E3C AC                       lodsb  ; Reserved
```

```
11484 00003E3D BF[F0430000]                mov   edi, rsvd15
11485 00003E42 E830FFFFFF                  call  write_hex
11486                                      ;
11487                                      ; (extension, not in original bios function)
11488 00003E47 66AD                        lodsw  ; I/O Port Base Address
11489 00003E49 BF[0D440000]                mov   edi, bPortAddr
11490 00003E4E E80EFFFFFF                  call  write_dhex
11491 00003E53 66AD                        lodsw  ; Control Port Address
11492 00003E55 BF[2A440000]                mov   edi, cPortAddr
11493 00003E5A E802FFFFFF                  call  write_dhex
11494 00003E5F AC                          lodsb  ; Head Register Upper Nibble
11495 00003E60 BF[47440000]                mov   edi, hregupnib
11496 00003E65 E80DFFFFFF                  call  write_hex
11497                                      ;
11498                                      ;;mov  ax, cs
11499                                      ;;mov  ds, ax
11500                                        ;sub ebx, ebx
11501 00003E6A A0[99460000]                mov    al, [ds_drv]
11502 00003E6F 88C3                        mov   bl, al
11503 00003E71 0430                        add   al, '0'
11504 00003E73 A2[87420000]                  mov    [dsknum], al
11505 00003E78 28FF                        sub   bh, bh
11506 00003E7A C0E302                      shl   bl, 2
11507 00003E7D 89DE                        mov   esi, ebx
11508 00003E7F 81C6[EA480000]              add   esi, disk_size
11509 00003E85 668B4602                    mov   ax, [esi+2]
11510 00003E89 BF[64440000]                mov   edi, disksize
11511 00003E8E E8CEFEFFFF                  call  write_dhex
11512 00003E93 668B06                      mov   ax, [esi]
11513 00003E96 BF[68440000]                mov   edi, disksize+4
11514 00003E9B E8C1FEFFFF                  call  write_dhex
11515                                      ;
11516 00003EA0 BE[7F420000]                  mov    esi, HDPT
11517 00003EA5 E8E7000000                  call  print_msg
11518 00003EAA C3                          retn
11519
11520                                  print_thdpt:
11521                                      ; 16/02/2015 (unix386.s)
11522                                      ; 25/12/2014 (dsectrm2.s)
11523                                      ; Writing the Translated FDPT on screen
11524                                      ; (PHOENIX - EDD specification v1.1)
11525 00003EAB 66AD                        lodsw  ; Logical Numbers of Cylinders, Limit 1024
11526 00003EAD BF[BD440000]                mov   edi, lcylnum
11527 00003EB2 E8AAFEFFFF                  call  write_dhex
11528 00003EB7 AC                          lodsb  ; Logical Numbers of Heads, Limit 256
11529 00003EB8 BF[DA440000]                mov   edi, lheadnum
11530 00003EBD E8B5FEFFFF                  call  write_hex
11531 00003EC2 AC                          lodsb  ; A0h signature, indicates translated table
11532 00003EC3 BF[F5440000]                mov   edi, tsignum
```

```
11533 00003EC8 E8AAFEFFFF                call  write_hex
11534 00003ECD AC                        lodsb ; Physical Sectors per Track
11535 00003ECE BF[10450000]              mov   edi, tpsptnum
11536 00003ED3 E89FFEFFFF                call  write_hex
11537 00003ED8 66AD                      lodsw ; Precompensation (Obsolete)
11538 00003EDA BF[2B450000]              mov   edi, tpcompnum
11539 00003EDF E87DFEFFFF                call  write_dhex
11540 00003EE4 AC                        lodsb ; Reserved
11541 00003EE5 BF[54450000]              mov   edi, trsvd7
11542 00003EEA E888FEFFFF                call  write_hex
11543 00003EEF AC                        lodsb ; Drive Control Byte
11544 00003EF0 BF[6F450000]              mov   edi, tdcbnum
11545 00003EF5 E87DFEFFFF                call  write_hex
11546 00003EFA 66AD                      lodsw ; Physical Cylinders, limit 65536
11547 00003EFC BF[8A450000]              mov   edi, tpcylnum
11548 00003F01 E85BFEFFFF                call  write_dhex
11549 00003F06 AC                        lodsb ; Physical Heads, limit 16
11550 00003F07 BF[A7450000]              mov   edi, tpheadnum
11551 00003F0C E866FEFFFF                call  write_hex
11552 00003F11 66AD                      lodsw ; Landing Zone (Obsolete)
11553 00003F13 BF[C2450000]              mov   edi, tlzonenum
11554 00003F18 E844FEFFFF                call  write_dhex
11555 00003F1D AC                        lodsb ; Logical Sectors per Track, Limit 63
11556 00003F1E BF[EB450000]              mov   edi, lsptnum
11557 00003F23 E84FFEFFFF                call  write_hex
11558 00003F28 AC                        lodsb ; Checksum for translated FDPT
11559 00003F29 BF[06460000]              mov   edi, checksum
11560 00003F2E E844FEFFFF                call  write_hex
11561                                    ;
11562                                    ; (extension, not in original bios function)
11563 00003F33 66AD                      lodsw ; I/O Port Base Address
11564 00003F35 BF[23460000]              mov   edi, tbPortAddr
11565 00003F3A E822FEFFFF                call  write_dhex
11566 00003F3F 66AD                      lodsw ; Control Port Address
11567 00003F41 BF[40460000]              mov   edi, tcPortAddr
11568 00003F46 E816FEFFFF                call  write_dhex
11569 00003F4B AC                        lodsb ; Head Register Upper Nibble
11570 00003F4C BF[5D460000]              mov   edi, thregupnib
11571 00003F51 E821FEFFFF                call  write_hex
11572                                    ;
11573                                    ;;mov  ax, cs
11574                                    ;;mov  ds, ax
11575                                      ;sub ebx, ebx
11576 00003F56 A0[99460000]              mov    al, [ds_drv]
11577 00003F5B 88C3                      mov   bl, al
11578 00003F5D 0430                      add   al, '0'
11579 00003F5F A2[7A440000]                mov    [tdsknum], al
11580 00003F64 28FF                      sub   bh, bh
11581 00003F66 C0E302                    shl   bl, 2
```

```
11582 00003F69 89DE                     mov    esi, ebx
11583 00003F6B 81C6[EA480000]           add    esi, disk_size
11584 00003F71 668B4602                 mov    ax, [esi+2]
11585 00003F75 BF[7A460000]             mov    edi, tdisksize
11586 00003F7A E8E2FDFFFF               call   write_dhex
11587 00003F7F 668B06                   mov    ax, [esi]
11588 00003F82 BF[7E460000]             mov    edi, tdisksize+4
11589 00003F87 E8D5FDFFFF               call   write_dhex
11590                                   ;
11591 00003F8C BE[72440000]             mov     esi, THDPT
11592                                   ;call  print_msg
11593                                   ;retn
11594
11595                                 print_msg:
11596                                   ;mov   bx, 7
11597                                     ;mov    ah, 0Eh
11598                                 pmsg_loop:
11599 00003F91 AC                       lodsb
11600 00003F92 20C0                     and    al, al
11601 00003F94 740D                     jz     short pmsg_ok
11602                                   ;int   10h
11603                                   ; 16/02/2015
11604 00003F96 56                       push   esi
11605 00003F97 31DB                     xor    ebx, ebx ; video page 0
11606 00003F99 B407                       mov     ah,  07h ; Black background, light gray forecolor
11607 00003F9B E89CD9FFFF               call   write_tty
11608 00003FA0 5E                       pop    esi
11609 00003FA1 EBEE                     jmp    short pmsg_loop
11610                                 pmsg_ok:
11611                                   ;mov   ah, 10h ; Getchar
11612                                   ;int   16h
11613 00003FA3 E82DF9FFFF               call   getch ; 16/02/2015
11614 00003FA8 C3                       retn
11615                                 ;
11616                                 FLPDPT:
11617 00003FA9 07                       db 07h
11618 00003FAA 0D0A                     db 0Dh, 0Ah
11619 00003FAC 4469736B20               db 'Disk '
11620                                 flpdnum:
11621 00003FB1 58202D20                 db 'X - '
11622 00003FB5 4449534B4554544520-      db 'DISKETTE PARAMETER TABLE'
11623 00003FBE 504152414D45544552-
11624 00003FC7 205441424C45
11625 00003FCD 0D0A0D0A                 db 0Dh, 0Ah, 0DH, 0Ah
11626 00003FD1 547970652020202020-      db 'Type                : '
11627 00003FDA 202020202020202020-
11628 00003FE3 2020203A20
11629                                 flpdtype:
11630 00003FE8 58202020                 db 'X   '
```

```
11631 00003FEC 5B2031203D20333630-          db '[ 1 = 360K, 2 = 1.2M, 3 = 720K, 4 = 1.44M ]'
11632 00003FF5 4B2C2032203D20312E-
11633 00003FFE 324D2C2033203D2037-
11634 00004007 32304B2C2034203D20-
11635 00004010 312E34344D205D
11636 00004017 0D0A0D0A                     db 0Dh, 0Ah, 0DH, 0Ah
11637 0000401B 535254202D20486561-          db 'SRT - Head Unld Time : '
11638 00004024 6420556E6C64205469-
11639 0000402D 6D65203A20
11640                                  rSrtHdUnld:
11641 00004032 585868202862697473-          db 'XXh (bits 0-3: SRT, bits 4-7: head unload time)'
11642 0000403B 20302D333A20535254-
11643 00004044 2C206269747320342D-
11644 0000404D 373A20686561642075-
11645 00004056 6E6C6F61642074696D-
11646 0000405F 6529
11647 00004061 0D0A                         db 0Dh, 0Ah
11648 00004063 444D41202D20486561-          db 'DMA - Head Load Time : '
11649 0000406C 64204C6F6164205469-
11650 00004075 6D65203A20
11651                                  rDmaHdLd:
11652 0000407A 585868202862697420-          db 'XXh (bit 0: 1 = DMA, bits 2-7: head load time)'
11653 00004083 303A2031203D20444D-
11654 0000408C 412C20626974732032-
11655 00004095 2D373A206865616420-
11656 0000409E 6C6F61642074696D65-
11657 000040A7 29
11658 000040A8 0D0A                         db 0Dh, 0Ah
11659 000040AA 4D6F746F72204F6666-          db 'Motor Off Count      : '
11660 000040B3 20436F756E74202020-
11661 000040BC 2020203A20
11662                                  bMotorOff:
11663 000040C1 585868202877697468-          db 'XXh (with 55ms icrements before turning off)'
11664 000040CA 2035356D7320696372-
11665 000040D3 656D656E7473206265-
11666 000040DC 666F7265207475726E-
11667 000040E5 696E67206F666629
11668 000040ED 0D0A                         db 0Dh, 0Ah
11669 000040EF 536563746F72205369-          db 'Sector Size          : '
11670 000040F8 7A6520202020202020-
11671 00004101 2020203A20
11672                                  bSectSize:
11673 00004106 585868202832203D20-          db 'XXh (2 = 512 bytes)'
11674 0000410F 353132206279746573-
11675 00004118 29
11676 00004119 0D0A                         db 0Dh, 0Ah
11677 0000411B 4C6173742053656374-          db 'Last Sect on a Track : '
11678 00004124 206F6E206120547261-
11679 0000412D 636B203A20
```

```
11680                                   bLastTrack:
11681 00004132 585868                         db 'XXh'
11682 00004135 0D0A                           db 0Dh, 0Ah
11683 00004137 476170204C656E6774-            db 'Gap Length  (R/W)    : '
11684 00004140 68202028522F572920-
11685 00004149 2020203A20
11686                                   bGapLen:
11687 0000414E 585868                         db 'XXh'
11688 00004151 0D0A                           db 0Dh, 0Ah
11689 00004153 44617461205472616E-            db 'Data Transfer Length : '
11690 0000415C 73666572204C656E67-
11691 00004165 7468203A20
11692                                   bDTL:
11693 0000416A 585868                         db 'XXh'
11694 0000416D 0D0A                           db 0Dh, 0Ah
11695 0000416F 476170204C656E6774-            db 'Gap Length (Format)  : '
11696 00004178 682028466F726D6174-
11697 00004181 2920203A20
11698                                   bGapFmt:
11699 00004186 585868                         db 'XXh'
11700 00004189 0D0A                           db 0Dh, 0Ah
11701 0000418B 46696C6C2043686172-            db 'Fill Char for format : '
11702 00004194 20666F7220666F726D-
11703 0000419D 6174203A20
11704                                   bFillChar:
11705 000041A2 58586820286E6F726D-            db 'XXh (normally F6h)'
11706 000041AB 616C6C792046366829
11707 000041B4 0D0A                           db 0Dh, 0Ah
11708 000041B6 486561642053657474-            db 'Head Settle Time     : '
11709 000041BF 6C652054696D652020-
11710 000041C8 2020203A20
11711                                   bHdSettle:
11712 000041CD 585868206D696C6C69-            db 'XXh milliseconds'
11713 000041D6 7365636F6E6473
11714 000041DD 0D0A                           db 0Dh, 0Ah
11715 000041DF 4D6F746F7220537461-            db 'Motor Startup Time   : '
11716 000041E8 727475702054696D65-
11717 000041F1 2020203A20
11718                                   bMotorOn:
11719 000041F6 5858682028696E2031-            db 'XXh (in 1/8th second intervals)'
11720 000041FF 2F387468207365636F-
11721 00004208 6E6420696E74657276-
11722 00004211 616C7329
11723 00004215 0D0A                           db 0Dh, 0Ah
11724                                         ; 19/12/2014
11725 00004217 0D0A                           db 0Dh, 0Ah
11726 00004219 4D6178696D756D2054-            db 'Maximum Track Number : '
11727 00004222 7261636B204E756D62-
11728 0000422B 6572203A20
```

```
11729                                   bMaxTrack:
11730 00004230 585868                         db 'XXh'
11731 00004233 0D0A                           db 0Dh, 0Ah
11732 00004235 44617461205472616E-            db 'Data Transfer Rate   : '
11733 0000423E 736665722052617465-
11734 00004247 2020203A20
11735                                   bDataRate:
11736 0000424C 585868202830306820-            db 'XXh (00h = 500KBS, 40h = 300KBS, 80H = 250KBS)'
11737 00004255 3D203530304B42532C-
11738 0000425E 20343068203D203330-
11739 00004267 304B42532C20383048-
11740 00004270 203D203235304B4253-
11741 00004279 29
11742 0000427A 0D0A                           db 0Dh, 0Ah
11743 0000427C 0D0A00                         db 0Dh, 0Ah, 0
11744
11745                                   ; 23/12/2014
11746                                   HDPT:
11747 0000427F 07                             db 07h
11748 00004280 0D0A                           db 0Dh, 0Ah
11749 00004282 4469736B20                     db 'Disk '
11750                                   dsknum:
11751 00004287 58202D20                       db 'X - '
11752 0000428B 464958454420444953-            db 'FIXED DISK PARAMETER TABLE'
11753 00004294 4B20504152414D4554-
11754 0000429D 4552205441424C45
11755 000042A5 0D0A0D0A                       db 0Dh, 0Ah, 0DH, 0Ah
11756 000042A9 4E756D626572206F66-            db 'Number of Cylinders : '
11757 000042B2 2043796C696E646572-
11758 000042BB 73203A20
11759                                   cylnum:
11760 000042BF 5858585868                     db 'XXXXh'
11761 000042C4 0D0A                           db 0Dh, 0Ah
11762 000042C6 4E756D626572206F66-            db 'Number of Heads     : '
11763 000042CF 204865616473202020-
11764 000042D8 20203A20
11765                                   headnum:
11766 000042DC 585868                         db 'XXh'
11767 000042DF 0D0A                           db 0Dh, 0Ah
11768 000042E1 526573657276656420-            db 'Reserved           : '
11769 000042EA 2020202020202020-
11770 000042F3 20203A20
11771                                   rsvd3:
11772 000042F7 585868                         db 'XXh'
11773 000042FA 0D0A                           db 0Dh, 0Ah
11774 000042FC 526573657276656420-            db 'Reserved           : '
11775 00004305 2020202020202020-
11776 0000430E 20203A20
11777                                   rsvd4:
```

```
11778 00004312 585868                      db 'XXh'
11779 00004315 0D0A                        db 0Dh, 0Ah
11780 00004317 507265636F6D70656E-         db 'Precompensation     : '
11781 00004320 736174696F6E202020-
11782 00004329 20203A20
11783                                  pcompnum:
11784 0000432D 5858585868                  db 'XXXXh'
11785 00004332 0D0A                        db 0Dh, 0Ah
11786 00004334 526573657276656420-         db 'Reserved            : '
11787 0000433D 2020202020202020-
11788 00004346 20203A20
11789                                  rsvd7:
11790 0000434A 585868                      db 'XXh'
11791 0000434D 0D0A                        db 0Dh, 0Ah
11792 0000434F 447269766520436F6E-         db 'Drive Control Byte  : '
11793 00004358 74726F6C2042797465-
11794 00004361 20203A20
11795                                  dcbnum:
11796 00004365 585868                      db 'XXh'
11797 00004368 0D0A                        db 0Dh, 0Ah
11798 0000436A 526573657276656420-         db 'Reserved            : '
11799 00004373 2020202020202020-
11800 0000437C 20203A20
11801                                  rsvd9:
11802 00004380 5858585868                  db 'XXXXh'
11803 00004385 0D0A                        db 0Dh, 0Ah
11804 00004387 526573657276656420-         db 'Reserved            : '
11805 00004390 2020202020202020-
11806 00004399 20203A20
11807                                  rsvd11:
11808 0000439D 585868                      db 'XXh'
11809 000043A0 0D0A                        db 0Dh, 0Ah
11810 000043A2 4C616E64696E67205A-         db 'Landing Zone        : '
11811 000043AB 6F6E65202020202020-
11812 000043B4 20203A20
11813                                  lzonenum:
11814 000043B8 5858585868                  db 'XXXXh'
11815 000043BD 0D0A                        db 0Dh, 0Ah
11816 000043BF 536563746F72732070-         db 'Sectors per Track   : '
11817 000043C8 657220547261636B20-
11818 000043D1 20203A20
11819                                  psptnum:
11820 000043D5 585868                      db 'XXh'
11821 000043D8 0D0A                        db 0Dh, 0Ah
11822 000043DA 526573657276656420-         db 'Reserved            : '
11823 000043E3 2020202020202020-
11824 000043EC 20203A20
11825                                  rsvd15:
11826 000043F0 585868                      db 'XXh'
```

```
11827 000043F3 0D0A                     db 0Dh, 0Ah
11828 000043F5 0D0A                     db 0Dh, 0Ah
11829 000043F7 492F4F20506F727420-       db 'I/O Port Base Addr  : '
11830 00004400 426173652041646472-
11831 00004409 20203A20
11832                                 bPortAddr:
11833 0000440D 5858585868               db 'XXXXh'
11834 00004412 0D0A                     db 0Dh, 0Ah
11835 00004414 436F6E74726F6C2050-       db 'Control Port Addr   : '
11836 0000441D 6F7274204164647220-
11837 00004426 20203A20
11838                                 cPortAddr:
11839 0000442A 5858585868               db 'XXXXh'
11840 0000442F 0D0A                     db 0Dh, 0Ah
11841 00004431 486561642052656720-       db 'Head Reg Upp Nibb   : '
11842 0000443A 557070204E69626220-
11843 00004443 20203A20
11844                                 hregupnib:
11845 00004447 585868                   db 'XXh'
11846 0000444A 0D0A                     db 0Dh, 0Ah
11847 0000444C 0D0A                     db 0Dh, 0Ah
11848 0000444E 53697A652028696E20-       db 'Size (in sectors)   : '
11849 00004457 736563746F72732920-
11850 00004460 20203A20
11851                                 disksize:
11852 00004464 585858585858585868       db 'XXXXXXXXh'
11853 0000446D 0D0A                     db 0Dh, 0Ah
11854 0000446F 0D0A00                   db 0Dh, 0Ah, 0
11855
11856                                 THDPT:
11857 00004472 07                       db 07h
11858 00004473 0D0A                     db 0Dh, 0Ah
11859 00004475 4469736B20               db 'Disk '
11860                                 tdsknum:
11861 0000447A 58202D20                 db 'X - '
11862 0000447E 5452414E534C415445-       db 'TRANSLATED FIXED DISK PARAMETER TABLE'
11863 00004487 442046495845442044-
11864 00004490 49534B20504152414D-
11865 00004499 4554552205441424C-
11866 000044A2 45
11867 000044A3 0D0A0D0A                 db 0Dh, 0Ah, 0DH, 0Ah
11868 000044A7 4C6F676963616C2043-       db 'Logical Cylinders   : '
11869 000044B0 796C696E6465727320-
11870 000044B9 20203A20
11871                                 lcylnum:
11872 000044BD 5858585868               db 'XXXXh'
11873 000044C2 0D0A                     db 0Dh, 0Ah
11874 000044C4 4C6F676963616C2048-       db 'Logical Heads       : '
11875 000044CD 656164732020202020-
```

```
11876 000044D6 20203A20
11877                                  lheadnum:
11878 000044DA 585868                        db 'XXh'
11879 000044DD 0D0A                          db 0Dh, 0Ah
11880 000044DF 5369676E6174757265-           db 'Signature          : '
11881 000044E8 202020202020202020-
11882 000044F1 20203A20
11883                                  tsignum:
11884 000044F5 585868                        db 'XXh'
11885 000044F8 0D0A                          db 0Dh, 0Ah
11886 000044FA 506879205365632070-           db 'Phy Sec per Track   : '
11887 00004503 657220547261636B20-
11888 0000450C 20203A20
11889                                  tpsptnum:
11890 00004510 585868                        db 'XXh'
11891 00004513 0D0A                          db 0Dh, 0Ah
11892 00004515 507265636F6D70656E-           db 'Precompensation     : '
11893 0000451E 736174696F6E202020-
11894 00004527 20203A20
11895                                  tpcompnum:
11896 0000452B 58585858682020284F-           db 'XXXXh  (Obsolete)'
11897 00004534 62736F6C65746529
11898 0000453C 0D0A                          db 0Dh, 0Ah
11899 0000453E 526573657276656420-           db 'Reserved           : '
11900 00004547 202020202020202020-
11901 00004550 20203A20
11902                                  trsvd7:
11903 00004554 585868                        db 'XXh'
11904 00004557 0D0A                          db 0Dh, 0Ah
11905 00004559 447269766520436F6E-           db 'Drive Control Byte  : '
11906 00004562 74726F6C2042797465-
11907 0000456B 20203A20
11908                                  tdcbnum:
11909 0000456F 585868                        db 'XXh'
11910 00004572 0D0A                          db 0Dh, 0Ah
11911 00004574 506879736963616C20-           db 'Physical Cylinders  : '
11912 0000457D 43796C696E64657273-
11913 00004586 20203A20
11914                                  tpcylnum:
11915 0000458A 5858585868                    db 'XXXXh'
11916 0000458F 0D0A                          db 0Dh, 0Ah
11917 00004591 506879736963616C20-           db 'Physical Heads      : '
11918 0000459A 486561647320202020-
11919 000045A3 20203A20
11920                                  tpheadnum:
11921 000045A7 585868                        db 'XXh'
11922 000045AA 0D0A                          db 0Dh, 0Ah
11923 000045AC 4C616E64696E67205A-           db 'Landing Zone        : '
11924 000045B5 6F6E65202020202020-
```

```
11925 000045BE 20203A20
11926                                  tlzonenum:
11927 000045C2 58585858682020284F-           db 'XXXXh  (Obsolete)'
11928 000045CB 62736F6C65746529
11929 000045D3 0D0A                          db 0Dh, 0Ah
11930 000045D5 4C6F67696320536563-           db 'Logic Sec per Trk   : '
11931 000045DE 207065722054726B20-
11932 000045E7 20203A20
11933                                  lsptnum:
11934 000045EB 585868                        db 'XXh'
11935 000045EE 0D0A                          db 0Dh, 0Ah
11936 000045F0 436865636B73756D20-           db 'Checksum            : '
11937 000045F9 202020202020202020-
11938 00004602 20203A20
11939                                  checksum:
11940 00004606 585868                        db 'XXh'
11941 00004609 0D0A                          db 0Dh, 0Ah
11942 0000460B 0D0A                          db 0Dh, 0Ah
11943 0000460D 492F4F20506F727420-           db 'I/O Port Base Addr  : '
11944 00004616 426173652041646472-
11945 0000461F 20203A20
11946                                  tbPortAddr:
11947 00004623 5858585868                    db 'XXXXh'
11948 00004628 0D0A                          db 0Dh, 0Ah
11949 0000462A 436F6E74726F6C2050-           db 'Control Port Addr   : '
11950 00004633 6F7274204164647220-
11951 0000463C 20203A20
11952                                  tcPortAddr:
11953 00004640 5858585868                    db 'XXXXh'
11954 00004645 0D0A                          db 0Dh, 0Ah
11955 00004647 486561642052656720-           db 'Head Reg Upp Nibb   : '
11956 00004650 557070204E69626220-
11957 00004659 20203A20
11958                                  thregupnib:
11959 0000465D 585868                        db 'XXh'
11960 00004660 0D0A                          db 0Dh, 0Ah
11961 00004662 0D0A                          db 0Dh, 0Ah
11962 00004664 53697A652028696E20-           db 'Size (in sectors)   : '
11963 0000466D 736563746F72732920-
11964 00004676 20203A20
11965                                  tdisksize:
11966 0000467A 585858585858585868            db 'XXXXXXXXh'
11967 00004683 0D0A                          db 0Dh, 0Ah
11968 00004685 0D0A00                        db 0Dh, 0Ah, 0
11969
11970                                  hex_digits:
11971                                  hexchrs:
11972 00004688 303132333435363738-           db '0123456789ABCDEF'
11973 00004691 39414243444546
```

```
11974
11975                                   inds:
11976 00004698 00                             db 0
11977                                   ds_drv:
11978 00004699 FF                             db 0FFh ; Current drive (on display)
11979 0000469A 00                             db 0    ; Current half (0 or >0)
11980                                   ds_sec:
11981 0000469B 00000000                       dd 0   ; Current sector (on display), drv 0
11982 0000469F 00000000                       dd 0   ; Current sector (on display), drv 1
11983 000046A3 00000000                       dd 0   ; Current sector (on display), drv 2
11984 000046A7 00000000                       dd 0   ; Current sector (on display), drv 3
11985 000046AB 00000000                       dd 0   ; Current sector (on display), drv 4
11986 000046AF 00000000                       dd 0   ; Current sector (on display), drv 5
11987                                   paragr:
11988 000046B3 00                             db 0
11989
11990                                   drv_names:
11991 000046B4 666430206664312068-            db 'fd0 fd1 hd0 hd1 hd2 hd3 '
11992 000046BD 643020686431206864-
11993 000046C6 322068643320
11994
11995                                   dpheader:
11996 000046CC 204472697665203A20             db ' Drive : '
11997                                   drv_name:
11998 000046D5 3030302020                     db '000  '
11999 000046DA 536563746F72203A20             db  'Sector : '
12000                                   sector_num:
12001 000046E3 464646464646464668             db  'FFFFFFFFh'
12002 000046EC 00                               db 0
12003
12004                                   current_txtpos:
12005 000046ED 00000000                       dd 0
12006                                   txtposoff:
12007 000046F1 00                             db 0 ; txtpos offset for sector number input
12008                                   dscmd:
12009 000046F2 00                             db 0   ; 0 = change drive
12010                                                ; 1 = change sector
12011                                                ; 2 = display disk parameters
12012                                   sdline:
12013 000046F3 204279746520                   db ' Byte '
12014                                   sdline_1:
12015 000046F9 30303068                       db '000h'
12016 000046FD 202D2020                       db ' -  '
12017                                   sdline_2:
12018 00004701 303020303020303020-            db '00 00 00 00 00 00 00 00 '
12019 0000470A 303020303020303020-
12020 00004713 303020303020
12021 00004719 303020303020303020-            db '00 00 00 00 00 00 00 00 '
12022 00004722 303020303020303020-
```

```
12023 0000472B 303020303020
12024 00004731 20                       db ' '
12025                                sdline_3:
12026 00004732 2E2E2E2E2E2E2E2E2E-      db '................'
12027 0000473B 2E2E2E2E2E2E2E
12028 00004742 20                       db 20h
12029
12030                                dpfooter1:
12031 00004743 204631203D20436861-      db ' F1 = Change Drive  '
12032 0000474C 6E6765204472697665-
12033 00004755 2020
12034 00004757 486F6D65203D204669-      db 'Home = First Sector '
12035 00004760 72737420536563746F-
12036 00004769 7220
12037 0000476B 50675570203D205072-      db 'PgUp = Previous Sector '
12038 00004774 6576696F7573205365-
12039 0000477D 63746F7220
12040 00004782 455343203D20455849-      db 'ESC = EXIT'
12041 0000478B 54
12042 0000478C 00                       db 0
12043                                dpfooter2:
12044 0000478D 204632203D20436861-      db ' F2 = Change Sector '
12045 00004796 6E676520536563746F-
12046 0000479F 7220
12047 000047A1 456E64203D204C6173-      db 'End = Last Sector   '
12048 000047AA 7420536563746F7220-
12049 000047B3 2020
12050 000047B5 5067446F776E203D20-      db 'PgDown = Next Sector   '
12051 000047BE 4E6578742053656374-
12052 000047C7 6F72202020
12053 000047CC 454E544552203D2050-      db 'ENTER = Prv/Nxt'
12054 000047D5 72762F4E7874
12055 000047DB 00                       db 0
12056                                ibcp:
12057 000047DC 00                       db 0   ; input box - row position
12058 000047DD 00                       db 0   ; input box - column position
12059                                F1_ib:
12060 000047DE 10                       db 16  ; box width (columns)
12061 000047DF 03                       db 3   ; box height (rows)
12062 000047E0 01                       db 1   ; label offset (vertical)
12063 000047E1 01                       db 1   ; label offset (horizontal)
12064 000047E2 01                       db 1   ; text (input) size
12065 000047E3 4E                       db 4Eh ; box color
12066 000047E4 44726976653A20           db 'Drive: '  ; Label
12067 000047EB 00                       db 0
12068
12069                                F2_ib:
12070 000047EC 14                       db 20  ; box width (columns)
12071 000047ED 03                       db 3   ; box height (rows)
```

```
12072 000047EE 01                       db 1   ; label offset (vertical)
12073 000047EF 01                       db 1   ; label offset (horizontal)
12074 000047F0 08                       db 8   ; text (input) size
12075 000047F1 4E                       db 4Eh ; box color
12076 000047F2 536563746F72203A20       db 'Sector : '  ; Label
12077 000047FB 00                       db 0
12078
12079                                 dskr_err:
12080                                   ;db 33 ; box width (columns)
12081 000047FC 11                       db 17
12082 000047FD 03                       db 3   ; box height (rows)
12083 000047FE 01                       db 1   ; label offset (vertical)
12084 000047FF 01                       db 1   ; label offset (horizontal)
12085 00004800 00                       db 0   ; text (input) size
12086 00004801 4E                       db 4Eh ; box color
12087                                   ;db 'Drive not ready or read error !'  ; Label
12088 00004802 204572726F72203A20       db ' Error : '
12089                                 err_code_str:
12090 0000480B 303068202120             db '00h ! '
12091 00004811 00                       db 0
12092
12093                                 ; Additional functions, variables/pointers for
12094                                 ; Real Mode adaption (out of unix386.s) variables/pointers
12095
12096                                 ;28/11/2014
12097                                 ; (set_cpos in dsectrm2.s)
12098                                 set_cposn:
12099 00004812 28DB                     sub   bl, bl ; video page 0
12100 00004814 668B15[3C490000]         mov   dx, [cursor_posn] ; dh = row, dl = column
12101 0000481B E94AD2FFFF               jmp   set_cpos
12102
12103                                 align 2
12104
12105                                 ; 16/02/2015
12106 00004820 0000                  cursor_posb: dw 0
12107                                 ; 01/12/2014
12108 00004822 0000                  cursor_shp: dw 0
12109
12110                                 ;; 10/12/2014
12111                                 ;; 24/11/2014 (Retro UNIX 386 v1)
12112                                 ; Physical drive type & flags
12113                                 ;fd0_type:   db 0  ; floppy drive type
12114                                 ;fd1_type:   db 0  ; 4 = 1.44 Mb, 80 track, 3.5" (18 spt)
12115                                                    ; 6 = 2.88 Mb, 80 track, 3.5" (36 spt)
12116                                                    ; 3 = 720 Kb, 80 track, 3.5" (9 spt)
12117                                                    ; 2 = 1.2 Mb, 80 track, 5.25" (15 spt)
12118                                                    ; 1 = 360 Kb, 40 track, 5.25" (9 spt)
12119                                 ;hd0_type:   db 0  ; EDD status for hd0 (bit 7 = present flag)
12120                                 ;hd1_type:   db 0  ; EDD status for hd1 (bit 7 = present flag)
```

```
 12121                                  ;hd2_type:   db 0  ; EDD status for hd2 (bit 7 = present flag)
 12122                                  ;hd3_type:   db 0  ; EDD status for hd3 (bit 7 = present flag)
 12123                                                ; bit 0 - Fixed disk access subset supported
 12124                                                ; bit 1 - Drive locking and ejecting
 12125                                                ; bit 2 - Enhanced disk drive support
 12126                                                ; bit 3 = Reserved (64 bit EDD support)
 12127                                                ; (If bit 0 is '1' Retro UNIX 386 v1
 12128                                                ; will interpret it as 'LBA ready'!)
 12129                                  ; 01/12/2014
 12130 00004824 0000                    drv_status: dw 0  ; fd0, fd1 (FFh = failure, 80h = existing)
 12131 00004826 00000000                          dd 0  ; hd0, hd1 hd2, hd3 (FFh =failure)
 12132                                                  ;                  (80h - 87h = exiting)
 12133                                                  ;                  (bit 0 = 1 : LBA ready)
 12134                                  ; 10/12/2014
 12135                                  ;hdc: db 0    ; Hard (Fixed) disk count
 12136                                  ;drv: db 0    ; physical drive number (0, 1, 80h, 81h, 82h, 83h)
 12137
 12138                                  ; 23/12/2014 (dw)
 12139                                  ; 16/12/2014 (db)
 12140 0000482A 0000                    wait_count: dw 0  ; INT 08h timer ticks for disk functions
 12141
 12142                                  ; 02/12/2014
 12143 0000482C 00000000                prev_sec:  dd 0   ; previous sector (before reading)
 12144 00004830 00                      last_drv:  db 0      ; 25/12/2014 - physical drv num of last hard disk
 12145
 12146                                  ; 03/01/2015
 12147 00004831 00                      LBAMode: db 0
 12148
 12149                                  prg_msg:
 12150 00004832 0D0A07                        db 0Dh, 0Ah, 07h
 12151 00004835 446973706C61792044-           db 'Display Disk Sectors in 386 Protected Mode - Retro UNIX 386 v1 Disk I/O
test.'
 12152 0000483E 69736B20536563746F-
 12153 00004847 727320696E20333836-
 12154 00004850 2050726F7465637465-
 12155 00004859 64204D6F6465202D20-
 12156 00004862 526574726F20554E49-
 12157 0000486B 582033383620763120-
 12158 00004874 4469736B20492F4F20-
 12159 0000487D 746573742E
 12160 00004882 0D0A                          db 0Dh, 0Ah
 12161 00004884 6279204572646F6761-           db 'by Erdogan Tan  [28/02/2015]'
 12162 0000488D 6E2054616E20205B32-
 12163 00004896 382F30322F32303135-
 12164 0000489F 5D
 12165 000048A0 0D0A0D0A                      db 0Dh, 0Ah, 0Dh, 0Ah
 12166 000048A4 28507265737320616E-             db '(Press any key to continue...)'
 12167 000048AD 79206B657920746F20-
 12168 000048B6 636F6E74696E75652E-
```

```
12169 000048BF 2E2E29
12170 000048C2 0D0A00                      db 0Dh, 0Ah, 0
12171
12172                                  ; 16/12/2014
12173                                  ;drv_not_ready:
12174                                  ;     db 07h, 0Dh, 0Ah
12175                                  ;     db 'Drive not ready !'  ; Temporary
12176                                  ;     db 0Dh, 0Ah, 0
12177
12178 000048C5 90                       align 2
12179
12180 000048C6 000000000000000000-     cylinders :  dw 0, 0, 0, 0, 0, 0
12181 000048CF 000000
12182 000048D2 000000000000000000-     heads   :  dw 0, 0, 0, 0, 0, 0
12183 000048DB 000000
12184 000048DE 000000000000000000-     spt     :  dw 0, 0, 0, 0, 0, 0
12185 000048E7 000000
12186 000048EA 000000000000000000-     disk_size :  dd 0, 0, 0, 0, 0, 0
12187 000048F3 000000000000000000-
12188 000048FC 000000000000
12189
12190                                  disk_pkt:
12191 00004902 10                            db 16 ; Packet size
12192 00004903 00                            db 0
12193 00004904 01                            db 1  ; Transfer count (1 sector)
12194 00004905 00                            db 0
12195                                  tbuff:
12196 00004906 [D04A]                        dw buffer ; offset
12197 00004908 0000                          dw 0 ; segment
12198                                  lba:
12199 0000490A 00000000                      dd 0
12200 0000490E 00000000                      dd 0
12201
12202                                  ; End of DSECTPM (Display Disk Sectors in Protected Mode) code
12203
12204                                  ; ////////////////////////////////////////////////////////////////////////////
12205
12206
12207                                  Align 2
12208
12209                                  ; 12/11/2014 (Retro UNIX 386 v1)
12210 00004912 00                     boot_drv:    db 0  ; boot drive number (physical)
12211                                  ; 24/11/2014
12212 00004913 00                     drv:       db 0  ; last drive number for hdd
12213 00004914 00                     hdc:       db 0  ; number of hard disk drives
12214                                                  ; (present/detected)
12215 00004915 90                     Align 2
12216                                  ;
12217                                  ; 24/11/2014 (Retro UNIX 386 v1)
```

```
12218                                   ; Physical drive type & flags
12219 00004916 00                       fd0_type:    db 0  ; floppy drive type
12220 00004917 00                       fd1_type:    db 0  ; 4 = 1.44 Mb, 80 track, 3.5" (18 spt)
12221                                                    ; 6 = 2.88 Mb, 80 track, 3.5" (36 spt)
12222                                                    ; 3 = 720 Kb, 80 track, 3.5" (9 spt)
12223                                                    ; 2 = 1.2 Mb, 80 track, 5.25" (15 spt)
12224                                                    ; 1 = 360 Kb, 40 track, 5.25" (9 spt)
12225 00004918 00                       hd0_type:    db 0  ; EDD status for hd0 (bit 7 = present flag)
12226 00004919 00                       hd1_type:    db 0  ; EDD status for hd1 (bit 7 = present flag)
12227 0000491A 00                       hd2_type:    db 0  ; EDD status for hd2 (bit 7 = present flag)
12228 0000491B 00                       hd3_type:    db 0  ; EDD status for hd3 (bit 7 = present flag)
12229                                                    ; bit 0 - Fixed disk access subset supported
12230                                                    ; bit 1 - Drive locking and ejecting
12231                                                    ; bit 2 - Enhanced disk drive support
12232                                                    ; bit 3 = Reserved (64 bit EDD support)
12233                                                    ; (If bit 0 is '1' Retro UNIX 386 v1
12234                                                    ; will interpret it as 'LBA ready'!)
12235                                   ;
12236                                   ; 04/11/2014 (Retro UNIX 386 v1)
12237 0000491C 0000                    mem_1m_1k:   dw 0 ; Number of contiguous KB between
12238                                                  ;   1 and 16 MB, max. 3C00h = 15 MB.
12239 0000491E 0000                    mem_16m_64k: dw 0 ; Number of contiguous 64 KB blocks
12240                                                  ;   between 16 MB and 4 GB.
12241 00004920 00000000                k_page_dir:  dd 0 ; Kernel's (System) Page Directory address
12242                                                  ;   (Physical address = Virtual address)
12243 00004924 00000000                memory_size: dd 0 ; memory size in pages
12244 00004928 00000000                free_pages:  dd 0 ; number of free pages
12245 0000492C 00000000                next_page:   dd 0 ; offset value in M.A.T. for
12246                                                  ;   first free page search
12247 00004930 00000000                last_page:   dd    0 ; offset value in M.A.T. which
12248                                                  ;   next free page search will be
12249                                                  ; stopped after it. (end of M.A.T.)
12250 00004934 00000000                first_page:  dd 0 ;   offset value in M.A.T. which
12251                                                  ; first free page search
12252                                                  ;   will be started on it. (for user)
12253 00004938 0000                    mat_size:    dw 0 ; Memory Allocation Table size in pages
12254
12255
12256                                   ; 02/09/2014 (Retro UNIX 386 v1)
12257                                   ; 04/12/2013 (Retro UNIX 8086 v1)
12258 0000493A 0000                    CRT_START:   dw 0    ; starting address in regen buffer
12259                                                       ; NOTE: active page only
12260 0000493C 0000<rept>              cursor_posn: times 8 dw 0 ; cursor positions for video pages
12261                                  active_page:
12262 0000494C 00                      ptty:            db 0      ; current tty
12263 0000494D 00                      db 0
12264                                  ; 02/09/2014 (Retro UNIX 386 v1)
12265 0000494E 00                      crt_ulc : db 0 ; upper left column (for scroll)
12266 0000494F 00                              db 0 ; upper left row (for scroll)
```

```
12267
12268 00004950 4F                    crt_lrc : db 79 ; lower right column (for scroll)
12269 00004951 18                            db 24 ; lower right row (for scroll)
12270
12271                                ; 07/09/2014
12272 00004952 0000<rept>            ttychr: times 8 dw 0     ; Character buffer (multiscreen)
12273
12274                                ; 06/11/2014 (Temporary data)
12275                                ; Memory Information message
12276                                msg_memory_info:
12277 00004962 4D454D4F525920414C-           db    "MEMORY ALLOCATION INFO", 0Dh, 0Ah, 0Dh, 0Ah
12278 0000496B 4C4F434154494F4E20-
12279 00004974 494E464F0D0A0D0A
12280 0000497C 546F74616C206D656D-           db    "Total memory  : "
12281 00004985 6F727920203A20
12282                                mem_total_b_str: ; 10 digits
12283 0000498C 303030303030303030-           db    "0000000000 bytes", 0Dh, 0Ah
12284 00004995 302062797465730D0A
12285 0000499E 2020202020202020-             db    "               ", 20h, 20h, 20h
12286 000049A7 2020202020202020-
12287 000049B0 20
12288                                mem_total_p_str: ; 7 digits
12289 000049B1 303030303030302070-           db    "0000000 pages", 0Dh, 0Ah
12290 000049BA 616765730D0A
12291 000049C0 0D0A                          db    0Dh, 0Ah
12292 000049C2 4D656D6F727920314D-           db    "Memory 1M-16M :   "
12293 000049CB 2D31364D203A202020
12294                                mem_1m_1k_str:   ; 5 digits
12295 000049D4 303030303020203148-           db    "00000  1K blocks", 0Dh, 0Ah
12296 000049DD 20626C6F636B730D0A
12297 000049E6 4D656D6F7279203136-           db    "Memory 16M-4G :   "
12298 000049EF 4D2D3447203A202020
12299                                mem_16m_64k_str: ; 5 digits
12300 000049F8 303030303020363448-           db      "00000 64K blocks", 0Dh, 0Ah
12301 00004A01 20626C6F636B730D0A
12302 00004A0A 0D0A                          db    0Dh, 0Ah
12303 00004A0C 46726565206D656D6F-           db    "Free memory : "
12304 00004A15 7279203A20
12305                                free_mem_b_str:  ; 10 digits
12306 00004A1A 303030303030303030-           db    "0000000000 bytes", 0Dh, 0Ah
12307 00004A23 302062797465730D0A
12308 00004A2C 2020202020202020-             db    "              ", 20h, 20h, 20h
12309 00004A35 2020202020202020
12310                                free_mem_p_str:  ; 7 digits
12311 00004A3D 303030303030302070-           db    "0000000 pages", 0Dh, 0Ah
12312 00004A46 616765730D0A
12313 00004A4C 0D0A                          db    0Dh, 0Ah
12314 00004A4E 466972737420667265-           db    "First free page : "
12315 00004A57 652070616765203A20
```

```
12316                                   next_mem_p_str:  ; 7 digits
12317 00004A60 303030303030300D0A                 db      "0000000", 0Dh, 0Ah
12318 00004A69 0D0A                             DB    0Dh, 0Ah
12319 00004A6B 4D2E412E542E207369-              db    "M.A.T. size: "
12320 00004A74 7A653A20
12321                                   mat_p_str:    ; 4 digits
12322 00004A78 303030302070616765-              db    "0000 pages", 0Dh, 0Ah
12323 00004A81 730D0A
12324 00004A84 4B65726E656C207061-              db    "Kernel page tables : "
12325 00004A8D 6765207461626C6573-
12326 00004A96 203A20
12327                                   pt_c_str:     ; 4 digits
12328 00004A99 30303030202B20310D-                db      "0000 + 1", 0Dh, 0Ah
12329 00004AA2 0A
12330 00004AA3 0D0A                             db    0Dh, 0Ah
12331 00004AA5 43616C63756C617465-              db    "Calculated free memory : "
12332 00004AAE 642066726565206D65-
12333 00004AB7 6D6F7279203A20
12334                                   free_mem_c_str:  ; 7 digits
12335 00004ABE 303030303030302070-              db    "0000000 pages", 0Dh, 0Ah
12336 00004AC7 616765730D0A
12337 00004ACD 0D0A00                           db    0Dh, 0Ah, 0
12338
12339                                   align 2
12340
12341                                   ; 12/02/2015
12342                                   buffer:
12343                                   ; 27/12/2013
12344                                   _end:  ; end of kernel code (and read only data, just before bss)
```