# CSCE 351
# Operating System Kernels

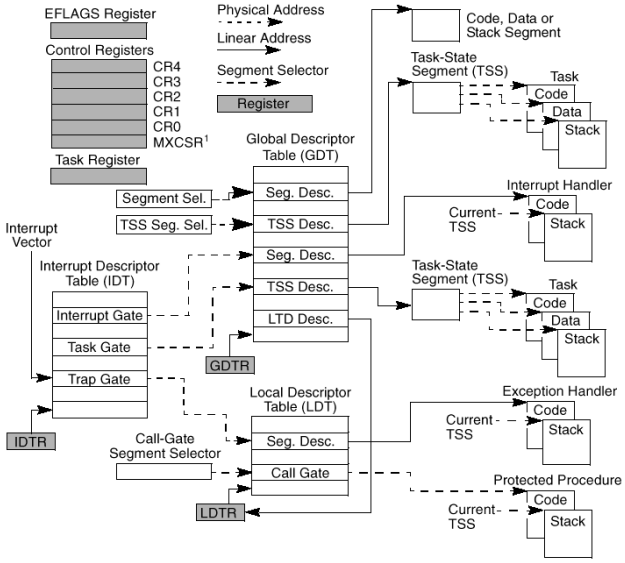
## Systems Programming for the Intel Architecture

Steve Goddard
*goddard@cse.unl.edu*

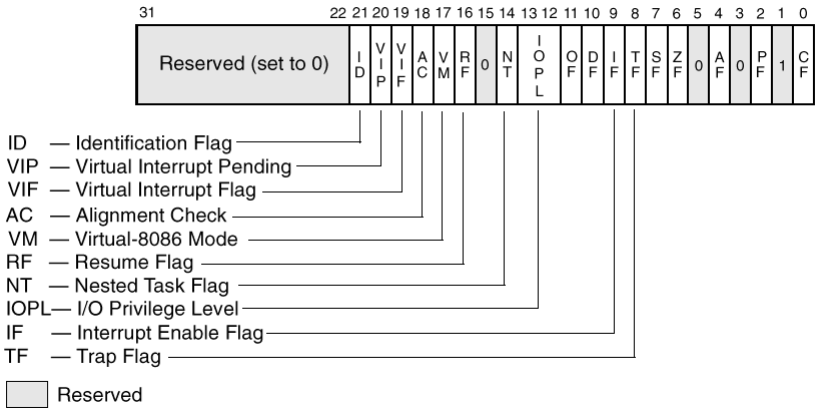
***http://www.cse.unl.edu/~goddard/Courses/CSCE351***



# System Level Registers and Data Structures

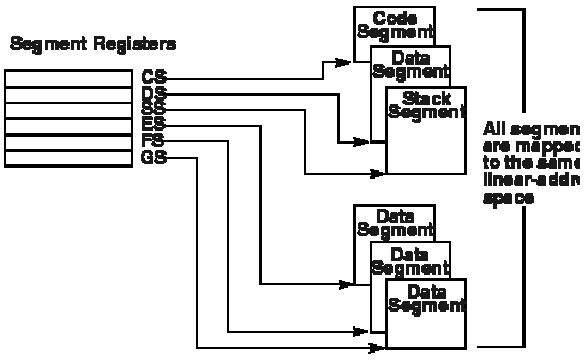


# System Flags in the EFLAGS Register

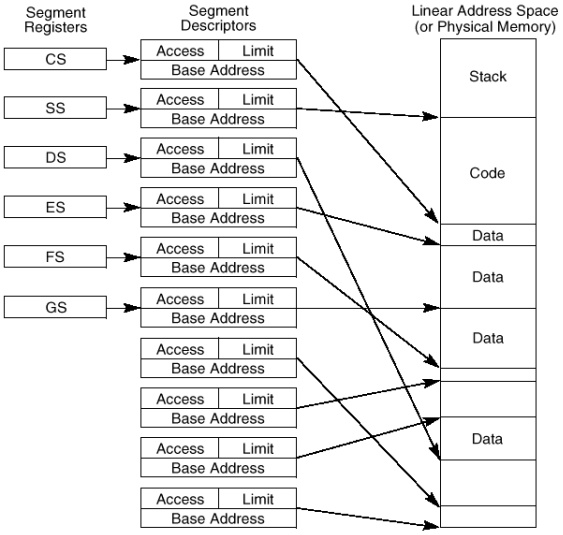





# Segmented Memory Model

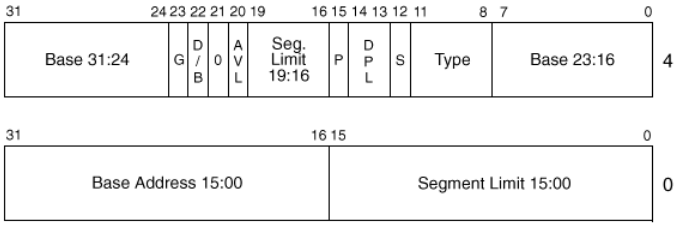


# Segmented Memory Model



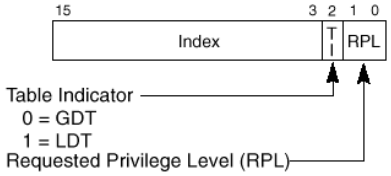



# Segment Descriptor



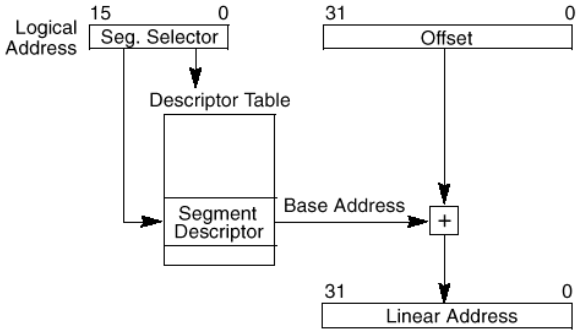

AVL — Available for use by system software
BASE — Segment base address
D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
DPL — Descriptor privilege level
G — Granularity
LIMIT — Segment Limit
P — Segment present
S — Descriptor type (0 = system; 1 = code or data)
TYPE — Segment type

# Segment Selector






# Logical Address to Linear Address Translation

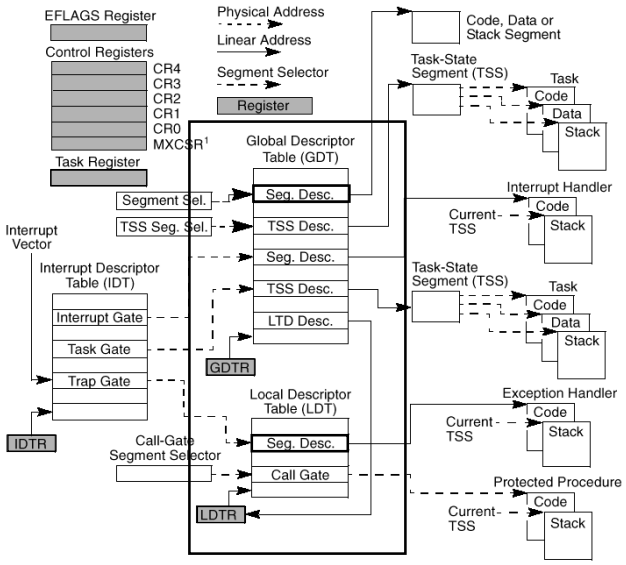


# Segment Registers

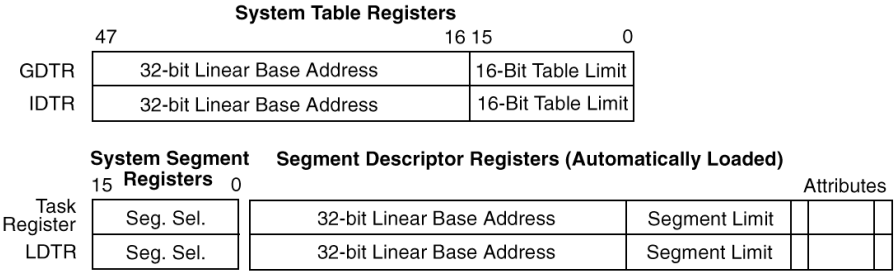
| Visible Part | Hidden Part | |
|---|---|---|
| Segment Selector | Base Address, Limit, Access Information | CS |
| | | SS |
| | | DS |
| | | ES |
| | | FS |
| | | GS |



# System Level Registers and Data Structures

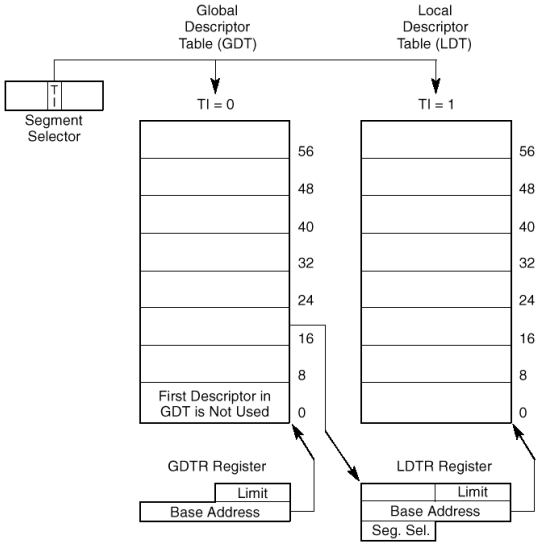


# Memory Management Registers



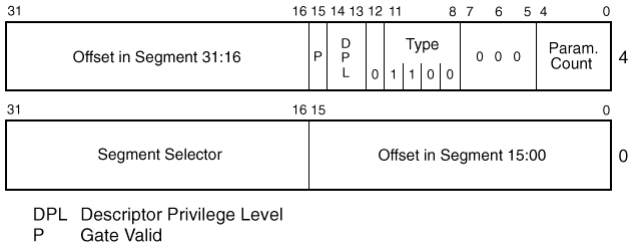



# Global Descriptor Table (GDT) and the GDT Register (GDTR)



# Gate Descriptors

- To provide controlled access to code segments with different privilege levels, the processor provides a special set of descriptors called gate descriptors. There are four kinds of gate descriptors:
  - » Call gates
  - » Trap gates
  - » Interrupt gates
  - » Task gates

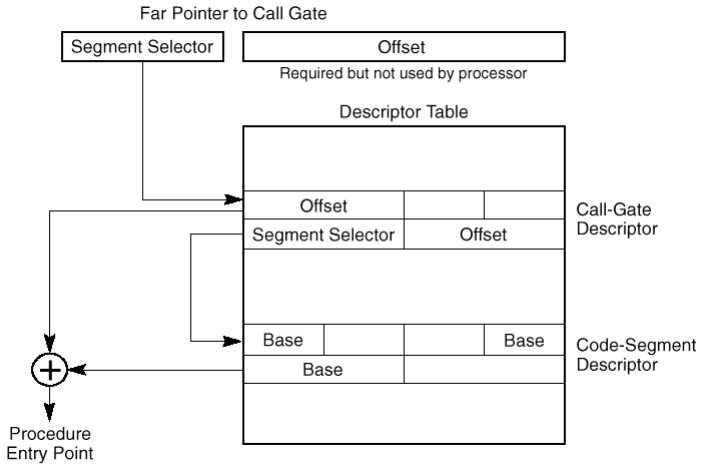





# Call Gates

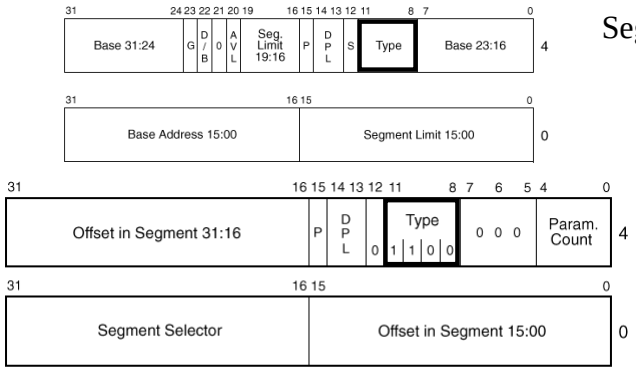
- A call-gate descriptor may reside in the GDT or in an LDT, but not in the interrupt descriptor table (IDT). It performs six functions:
  1. It specifies the code segment to be accessed.
  2. It defines an entry point for a procedure in the specified code segment.
  3. It specifies the privilege level required for a caller trying to access the procedure.
  4. If a stack switch occurs, it specifies the number of optional parameters to be copied between stacks.
  5. It defines the size of values to be pushed onto the target stack: 16-bit gates force 16-bit pushes and 32-bit gates force 32-bit pushes.
  6. It specifies whether the call-gate descriptor is valid.

# Call Gate Mechanism






# Segment Descriptor vs. Gate Descriptor



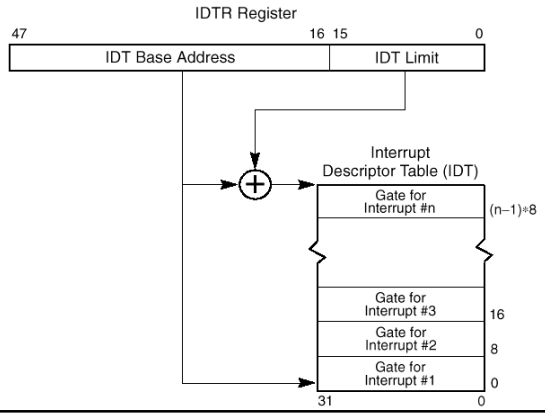

Segment Descriptor

Gate Descriptor

DPL Descriptor Privilege Level
P Gate Valid

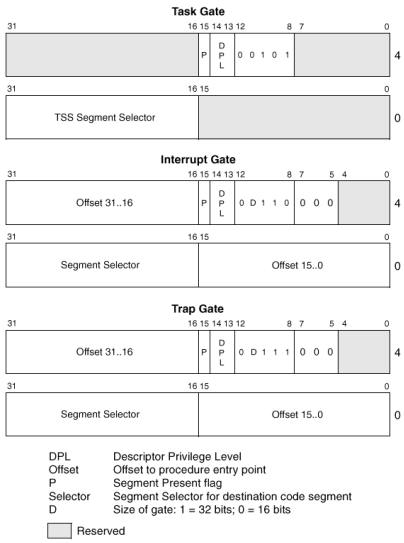
# Interrupt Descriptor Table (IDT)

- Associates each exception or interrupt vector with a gate descriptor for the procedure or task used to service the associated exception or interrupt.






# IDT Gate Descriptors



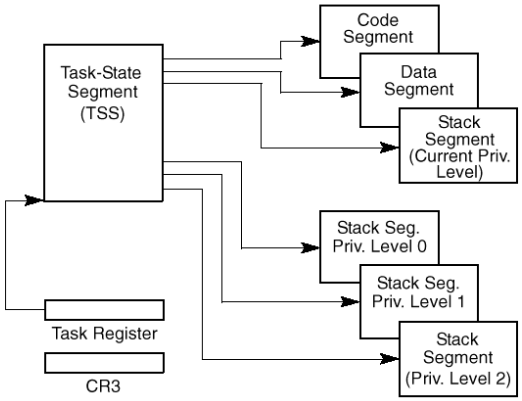
# Task Management

- The Intel Architecture provides a mechanism for
  - » saving the state of a task,
  - » for dispatching tasks for execution, and
  - » for switching from one task to another.
- When operating in protected mode, all processor execution takes place from within a task.
- A task is made up of two parts:
  - » a task execution space
  - » task-state segment (TSS).



# Task State Segment (TSS)

- The TSS specifies the segments that make up the task execution space and provides a storage place for task state information.

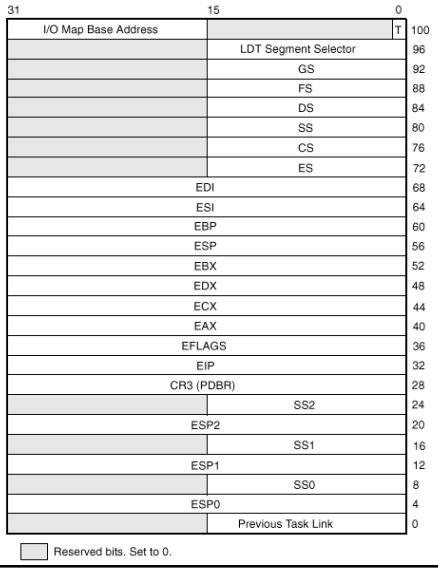


# Task State

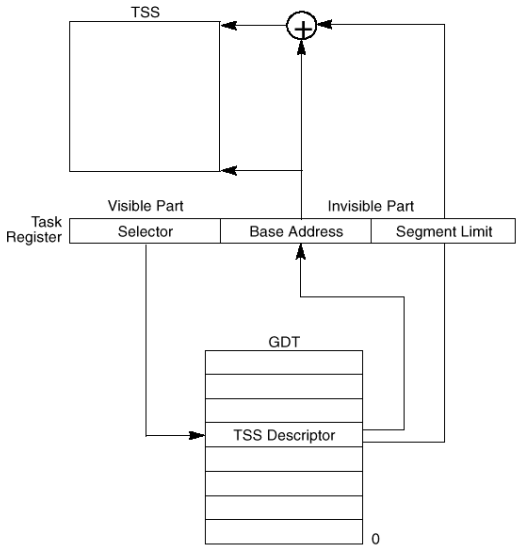
- The following items define the state of the currently executing task:
  - » The task's current execution space, defined by the segment selectors in the segment registers (CS, DS, SS, ES, FS, and GS).
  - » The state of the general-purpose registers.
  - » The state of the EFLAGS register.
  - » The state of the EIP register.
  - » The state of control register CR3.
  - » The state of the task register.
  - » The state of the LDTR register.
  - » The I/O map base address and I/O map (contained in the TSS).
  - » Stack pointers to the privilege 0, 1, and 2 stacks (contained in the TSS).
  - » Link to previously executed task (contained in the TSS).



# TSS Structure

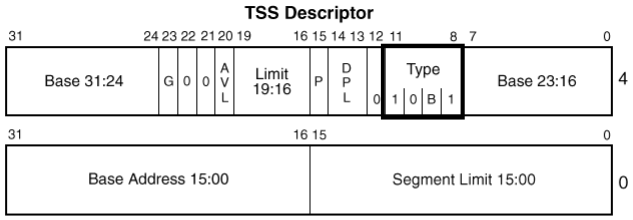


# Task Register



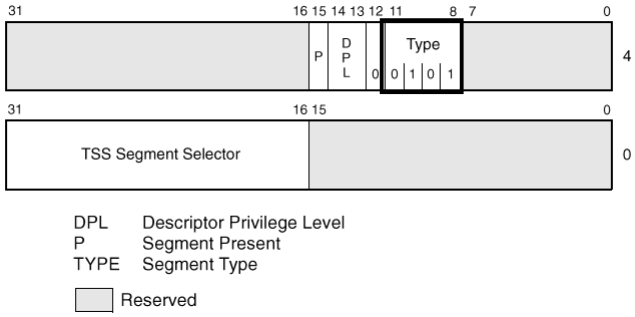



# TSS Descriptor




| | |
|---|---|
| AVL | Available for use by system software |
| B | Busy flag |
| BASE | Segment Base Address |
| DPL | Descriptor Privilege Level |
| G | Granularity |
| LIMIT | Segment Limit |
| P | Segment Present |
| TYPE | Segment Type |

# Task Gate Descriptor

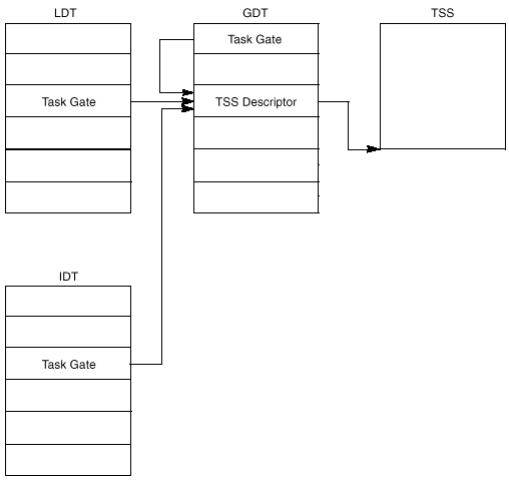



DPL Descriptor Privilege Level
P Segment Present
TYPE Segment Type

Reserved



# Task Gates Referencing the Same Task



# Putting It All Together