File Systems

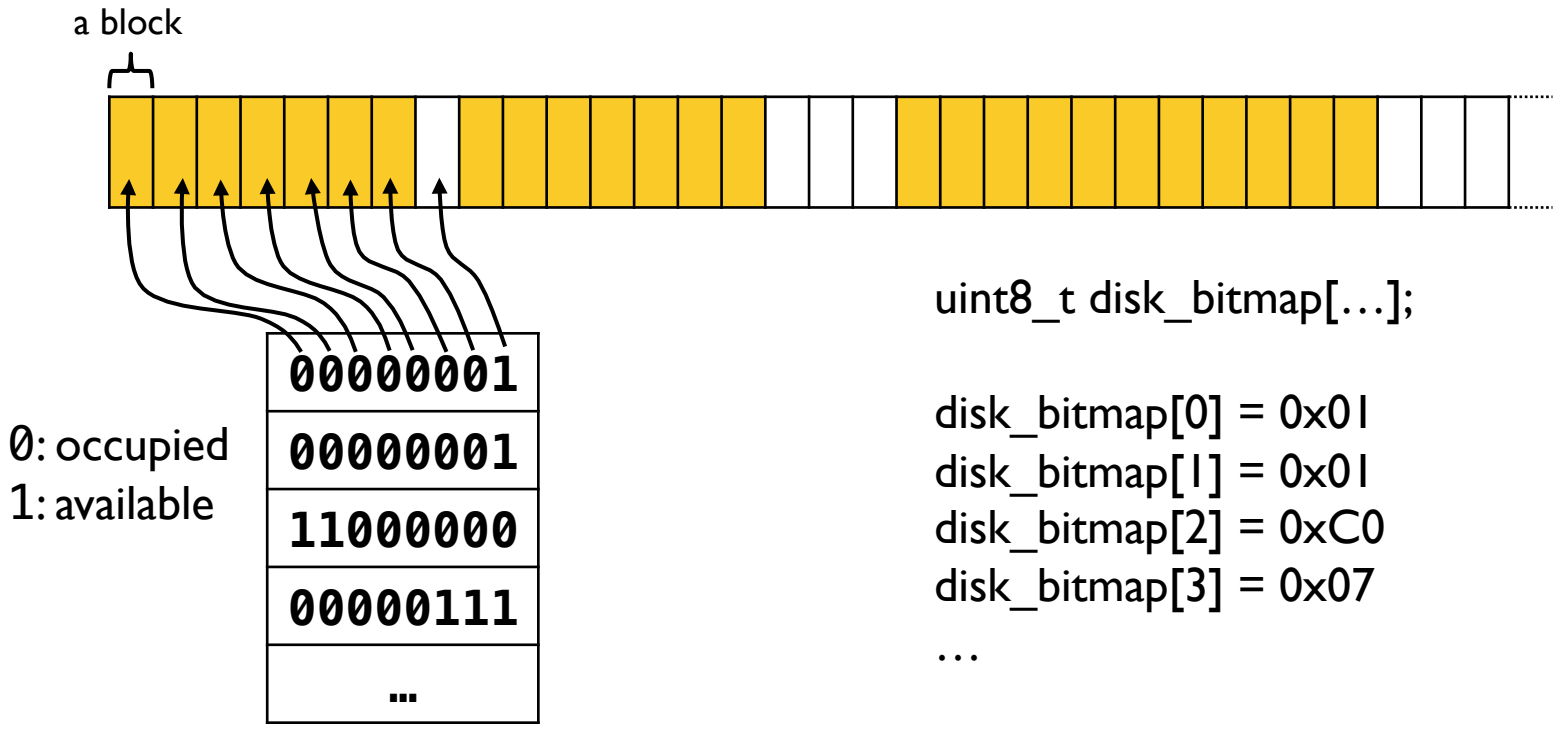
COMP 3361: Operating Systems I

Winter 2015

http://www.cs.du.edu/3361

1 Free Space Management

- Keeping track of free blocks on disk
- Using **bitmaps**




```
uint8_t disk_bitmap[…];

disk_bitmap[0] = 0x01
disk_bitmap[1] = 0x01
disk_bitmap[2] = 0xC0
disk_bitmap[3] = 0x07
…
```

2 Linked List Based Free Space

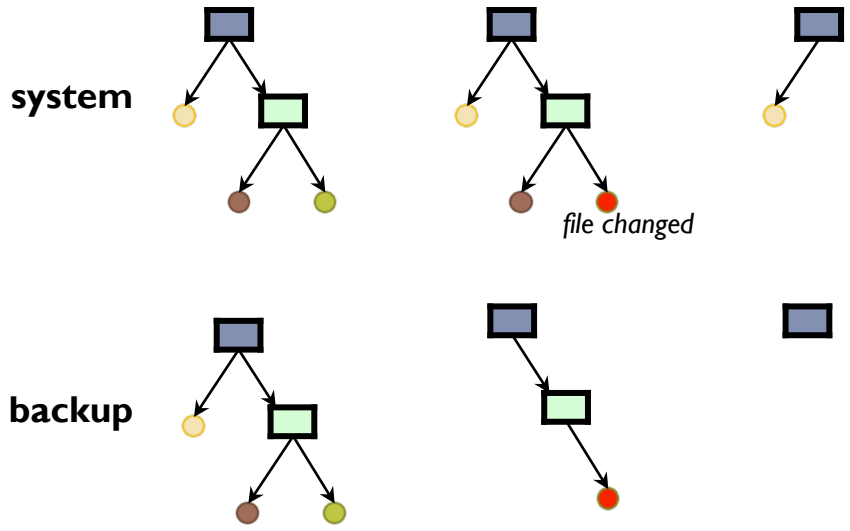
- Use a **free list**: store block numbers of all blocks that are free
  - reserve some blocks where these numbers will be stored
- If free blocks appear in long runs, store start free block number and then number of blocks free from there
  - not good when disk is heavily fragmented
- Only part of the free list can be in memory at a time

3 File System Backup

- **Physical**
  - backup each block of the disk
  - use free blocks and bad blocks information from file system
- **Logical**
  - backup selected files and directories
    - typically only user created data
- **Full**
  - backup entire target (full disk or selected parts)
- **Incremental**
  - backup only data that has changed since last backup

Incremental Backup

- First backup is always a full backup
- Thereafter, backup
  - file: if file has changed
  - directory: if any file, sub-directory, or files/directories inside sub-directories have changed



5 File System Consistency

- Check consistency at system boot
- **Block consistency**: a block must be either in the free list or part of one single file
  - restore missing blocks by adding to free list
  - remove duplication in free list
  - remove duplication in file metadata by copying block
- **Hard link consistency**: the number of times an i-node number appears in the directories must be equal to the link count stored in the i-node
  - restore link count

6 File System Performance

- Caching: keep copies of some blocks in memory
  - use replacement algorithms when cache is full
    - is block likely to be needed again soon?
    - is block essential for file system consistency?
  - synchronize periodically
    - called write-through cache if synchronized immediately
- Read blocks into cache ahead of time
  - performance improvement if file access is sequential and file blocks are contiguous
- Reduce disk-arm latency in mechanical hard drives
  - allocate blocks from same cylinder group to a file (no seek after first block)
  - keep i-nodes close to file blocks

7 References

- Chapter 4.4, Modern Operating Systems, A. Tanenbaum and H. Bos, 4th Edition.