More on Disk Storage and File System

CS 256/456

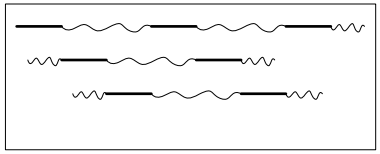
Dept. of Computer Science, University of Rochester

Recap of the Last Class: Disk Storage

- Disk drive
  - mechanical parts (cylinders, tracks, sectors) and how they move to access disk data
  - electronic part (disk controller main) exposes an one-dimensionally addressable set of blocks
  - large seek/rotation time
- Disk scheduling
  - goals: overall efficiency; fairness (no starvation)
  - FCFS, SSTF, SCAN, C-SCAN, LOOK

Concurrent I/O

- Consider multiple request handlers in a Web server
  - each accesses a different stream of sequential data (a file) on disk;
  - each reads a chunk (the buffer size) at a time; does a little CPU processing; and reads the next chunk
- What happens?



How to Deal with It?

- Aggressive prefetching
- Anticipatory scheduling [Iyer & Druschel, SOSP 2001]
  - at the completion of an I/O request, the disk scheduler will wait a bit (despite there is other work to do), in anticipation that a new request with strong locality will be issued. go ahead to schedule another request if no such new request appears before timeout.
  - included in Linux 2.6

Scheduling at Disk and OS

- Some disks are smart (support scheduling of its own)
  - SCSI vs. IDE disks
- Scheduling at disk vs. at OS
  - FCFS, SSTF, elevator, anticipatory
- How does an OS balance it?

File System Abstraction

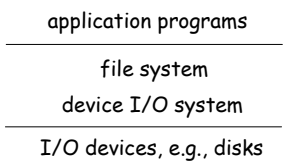
- File system is the OS abstraction for storage resources
  - File is a logical storage unit in the OS abstract interface for storage resources
  - Directory is a logical "container" for a group of files
- File attribute:
  - name, size, access time, sharing/protection, location
- File access:
  - sequential: open()/read()
  - random: seek()
- File structure (mostly concerns of user programs)
  - None - sequence of words, bytes
  - Complex Structures
    - records/formatted document/executable file

File Sharing and Protection

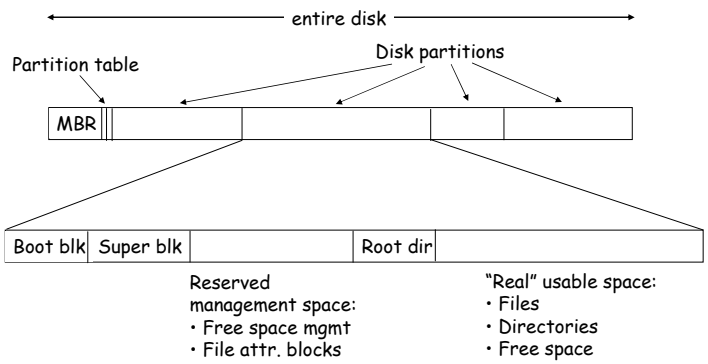
- Sharing of files on multi-user systems is desirable.
- Sharing must accompany a *protection* scheme.
  - In general, a protection scheme specifies whether any specific user can access any specific file.
  - What is it done right now?
  - Is it flexible enough?

File System Implementation

- Abstractions:
  - file: storage unit
  - directory: container of files
- OS responsibility for file management:
  - Manipulation of files and directories.
  - Map files onto (nonvolatile) secondary storage - disks.



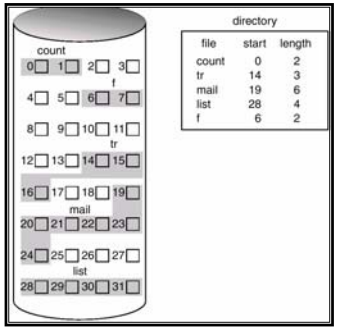
File System Layout



File on the Disk

- Disk basic allocation unit is a block (e.g., 512 bytes)
- File system may choose to use a larger block size (e.g., 4KB)
- File allocation methods
  - How disk blocks are allocated for files
  - Contiguous allocation, linked allocation, indexed allocation
  - Metrics: access speed (sequential & random), space utilization

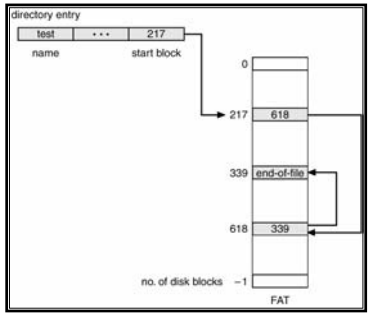
Contiguous File Allocation



- Each file occupies a set of contiguous blocks on the disk.
- Advantage:
  - Simple – only starting location (block #) and length (number of blocks) are required.
  - Fast sequential; also quite fast random access.
- Disadvantage:
  - External fragmentation
  - Inflexible when appending to a file

Linked File Allocation

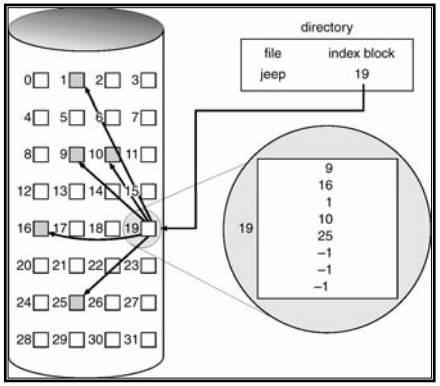
- Each file is a linked list of disk blocks
  - each block contains a next pointer
  - directory only needs to store the pointer to the first block
  - blocks may be scattered anywhere on the disk.



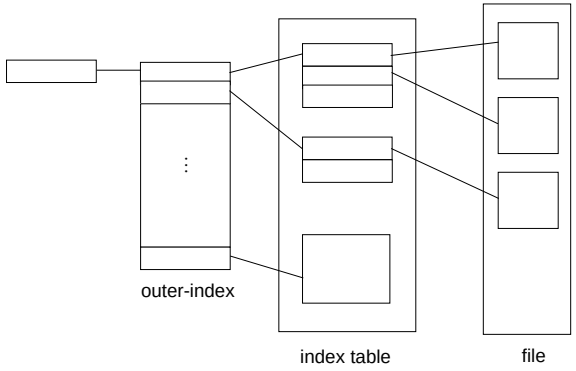
- Advantage
  - Space efficient
  - Flexible in appending
- Disadvantage:
  - Poor access speed (sequential & random)

Indexed File Allocation

- Brings all pointers together into the *index block.*



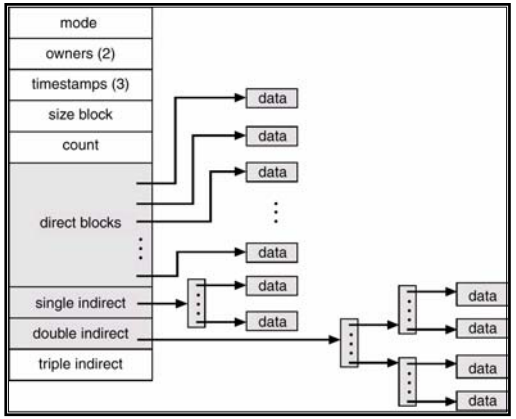
Multi-level Indexed File Allocation



Indexed Allocation (pros and cons)

- Space efficiency
  - no external fragmentation
  - overhead of index blocks
- Access speed
  - random access
  - sequential access

UNIX (4K bytes per block)



Disclaimer

- Parts of the lecture slides contain original work of Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Andrew S. Tanenbaum, and Gary Nutt. The slides are intended for the sole purpose of instruction of operating systems at the University of Rochester. All copyrighted materials belong to their original owner(s).