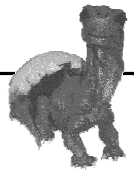
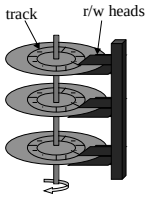


File Systems II

Comp 305 Lecture 9

How Does a Disk Work?

- To read/write disk sector
  - Move head to cylinder (seek)
  - Wait for sector to rotate to head (latency)
  - Read or write (transfer)
- Controller
  - One controller for several disks
  - Intelligent controllers
  - Cache



Issues in Disk and File System Performance

- Addressing
- Space Management
- Scheduling

Addressing

- Logical view is array [0..N-1] of blocks
- For a disk with

  s sectors per track and

  t tracks per cylinder

  we map cylinder *i*, surface *j*, and sector *k* to block *b* by:

  $b = k + s \times (j + i \times t)$

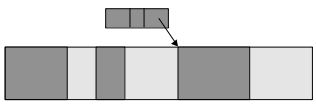
Space Management

- Allocation
  - Contiguous
  - Linked
  - Indexed
- Free space management

11001111
00011110
01101111
00011011

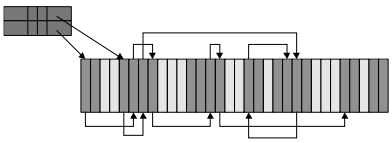
Contiguous Allocation

- File defined by base and length
- Sequential and direct access supported
- Difficult to allocate space or increase file size



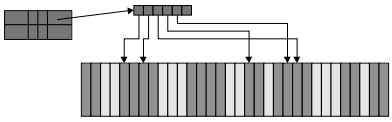
Linked Allocation

- File defined by first and last
- Solves storage problem - any free block will do
- Direct access not supported (efficiently)



Indexed Allocation

- File defined by index
- Supports sequential and direct access
- Solves storage problem with overhead of index blocks



Principles of Disk Scheduling

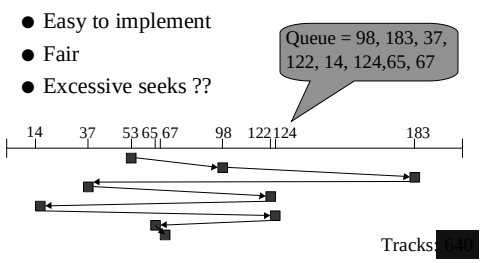
- Goals:
  - Increase throughput
  - Reduce mean response time
  - Control variance of response time
  - Minimal impact on system performance

Scheduling Algorithms

- FCFS - First come first served
- SSTF - Shortest seek time first
- SCAN
- N-step SCAN
- C-SCAN - Circular scan

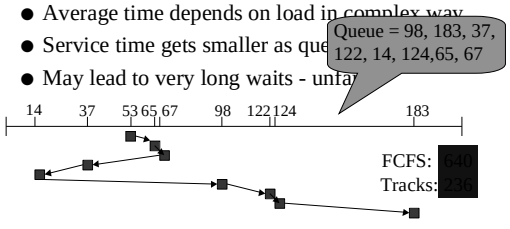
First Come First Served

- Easy to implement
- Fair
- Excessive seeks ??



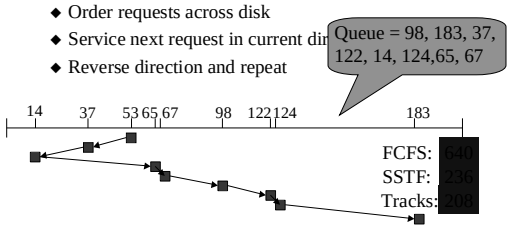
Shortest Seek Time First

- Minimises seek time
- Average time depends on load in complex way
- Service time gets smaller as que
- May lead to very long waits - unfa



SCAN

- SSTF in one direction at a time
  - Order requests across disk
  - Service next request in current dir
  - Reverse direction and repeat



Locality

- Previous results assume each cylinder is equally likely
- Different behaviour between a personal machine and a server
- Placement of files may improve locality

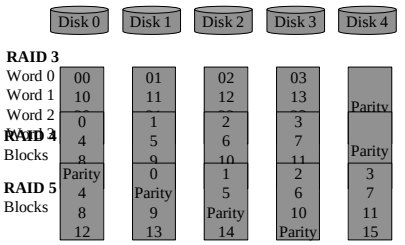
RAID

- Redundant Array of Inexpensive Disks
- Replace single large disk with lots of “mass produced” disks
- Increased parallelism
- Increased reliability
- Six levels

RAID Levels

Level 0 - Data striped across the disk
Level 1 - Mirrored disk array
Level 2 - Hamming-coded disk array
Level 3 - Single byte stripe with parity
Level 4 - Arbitrary stripe with parity
Level 5 - Distributed parity

RAID Levels 3, 4 & 5



RAID Performance

- At low loads
  - Level 0 performs best
  - No redundancy
- At high loads
  - Level 4 is worse as parity is not read
  - Level 4 has contention for parity on writes
  - Level 5 preferred

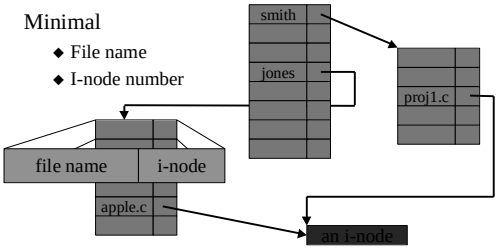
The Unix File System

- Directories
- I-nodes
- Disk organisation
- Open file structure
- NFS

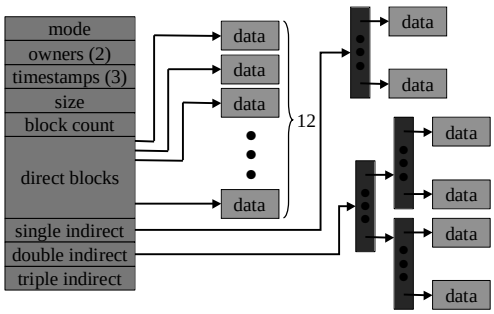
The Unix Directory

Minimal

- File name
- I-node number



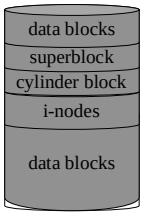
The I-node



Disk Organisation

- Disk may hold several file systems
- A file system resides on a logical disk
  - Boot block
  - Cylinder groups

Cylinder Group Layout



Blocks and Fragments

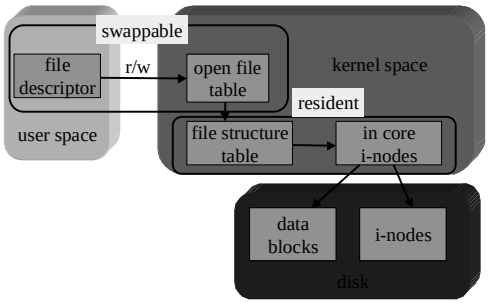
- Blocks are a multiple of the sector size
- Larger blocks make for more efficient transfers
- Smaller blocks make for more efficient storage
- Compromise on both

  block (4-8k)
  fragment (.5-2k)

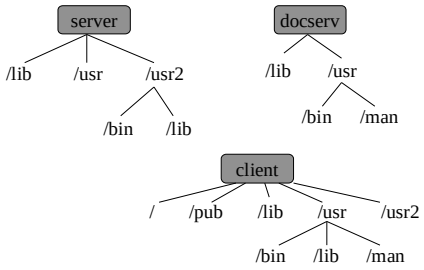
Allocation Policy

- Attempts to group related blocks
- Files from same cylinder group as directory
- Sub-directories go to a new cylinder group
- Indirect blocks from another cylinder group

Open File Structure



Network File System



NFS Architecture