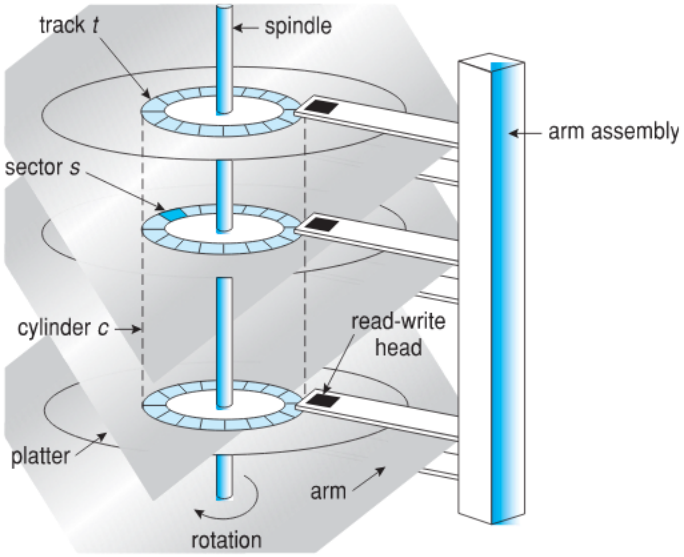
Mass Storage Systems - And Dinosaurs, boi

1 Overview of Mass Storage Structure

- Magnetic disks provide bulk of secondary storage of modern computers
  - Transfer rate is the rate at which data flows between the drive and the computer
  - Positioning time (random access time) is time to move disk arm to desired cylinder (seek time) and time for desired sector to rotate under the disk head (rotational latency)
  - Head crash results from a disk head making contact with the disk surface
- Disks can be removable
- Drive attached to computer via I/O bus
  - Busses vary
  - Host controller in computer uses bus to talk to disk controller built into drive or storage array

2 Moving head disk mechanism



3 Hard Disk Performance

- Access Latency = Average access time = Average seek time + average latency
- Average I/O time = average access time + (amount to transfer/transfer rate)+controller overhead

4 Solid State Disks

- Non-volatile memory used like a hard drive
- Can be more reliable than HDDs
- More expensive per MB
- Maybe have a shorter life span
- Less capacity
- But much faster
- Busses can be too slow → connect directly to PCI for example
- No moving parts, so no seek time or rotational latency

5 Magnetic Tape

- Relatively permanent and holds large quantities of data
- Access time slow
- Random access 1000 times slower than disk
- Mainly used for backup, storage of infrequently used data, transfer medium between systems
- Kept in spool and wound or rewound past read-write head

6 Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of logical blocks, where the logical block is the smallest unit of transfer
- The 1 dimensional array of logical blocks is mapped into the sectors of the disk sequentially
  - Sector 0 is the first sector of the first track on the outermost sylinder
  - Mapping proceeds in order though that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost
- Logical to physical addresses should be easy
  - Except for bad sectors
  - Non constant number of sectors per track via constant angular velocity

7 Storage Array

- Can just attach disks,or arrays of disks
- Storage array has controller(s), provides features to attached host(s)
  - Ports to connect hosts to array
  - Memory, controlling software
  - A few to thousands of disks
  - RAID, hot spares, hot swap
  - Shared storage → more efficiency
  - Features found in some file systems
    - Snapshots, clones, thin provisioning, replication, de duplication etc

8 Disk Scheduling

- The operating system is responsible for using hardware efficiently - for the disk drives, this means having a fast access time and disk bandwidth
- Minimize seek time
- Seek time ≈ seek distance
- Disk **bandwidth** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer
- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- I/O request includes input or output mode, disk address, memory address, number of sectors to transfer

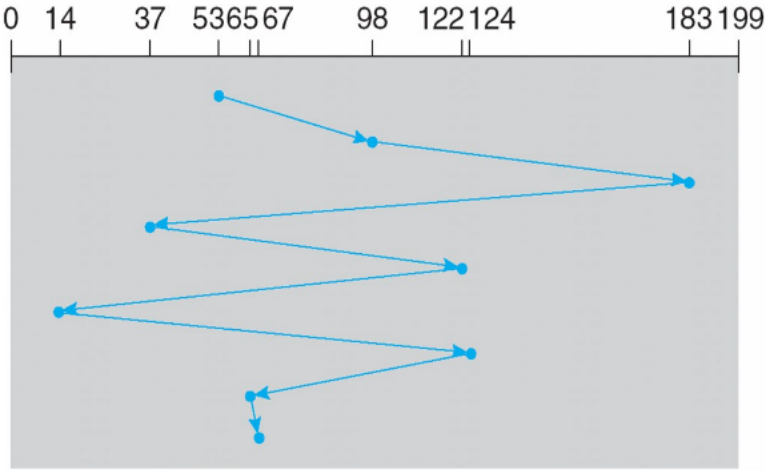
- OS maintains queue of request, per disk or device
- Idle disk can immediately work on I/O request, busy disk means work must queue
  - Optimisation algorithms only make sense when a queue exists
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying ”depth”)
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199)

$$98, 183, 37, 122, 14, 124, 65, 67$$

Head pointer 53

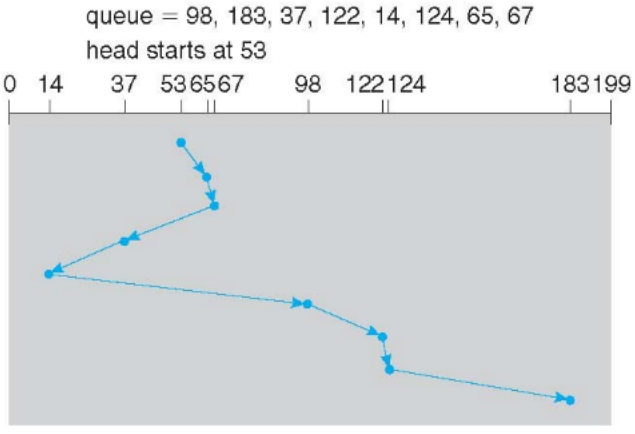
9 FCFS

Illustration shows total head movement of 640 cylinders
queue =98,183,37,122,14,124,65,67
Heads starts at 53



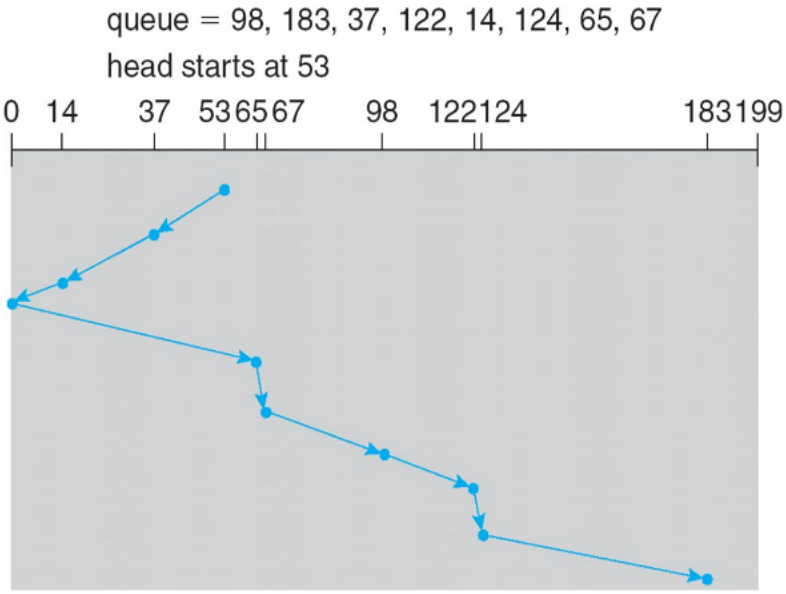
10 SSTF

- Shortest Seek time first selects the request with the minimum seek time from the current head position
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests
- Illustration shows total head movement of 236 cylinders



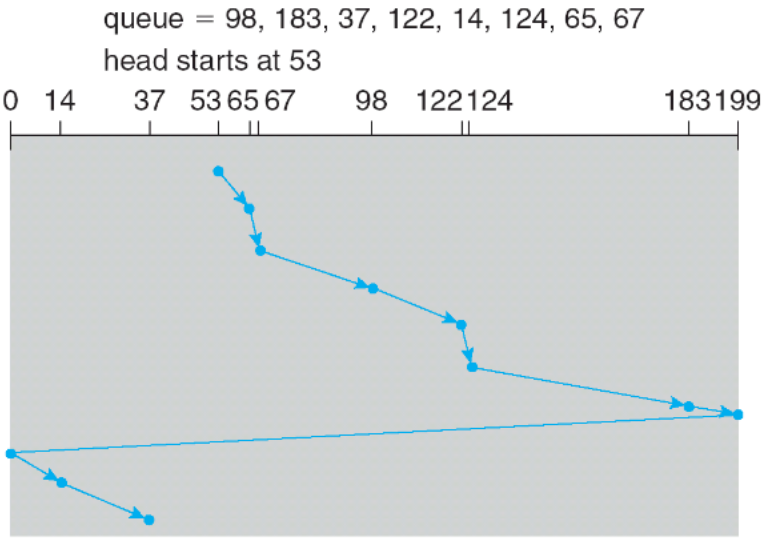
11 SCAN

- The disk arm starts at one end of the disk, and moves towards the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues
- SCAN algorithm, sometimes called the elevator algorithm
- Illustration shows total head movement of 236 cylinders
- But note that if the requests are uniformly dense, largest density at other end of disk and those wait the longest



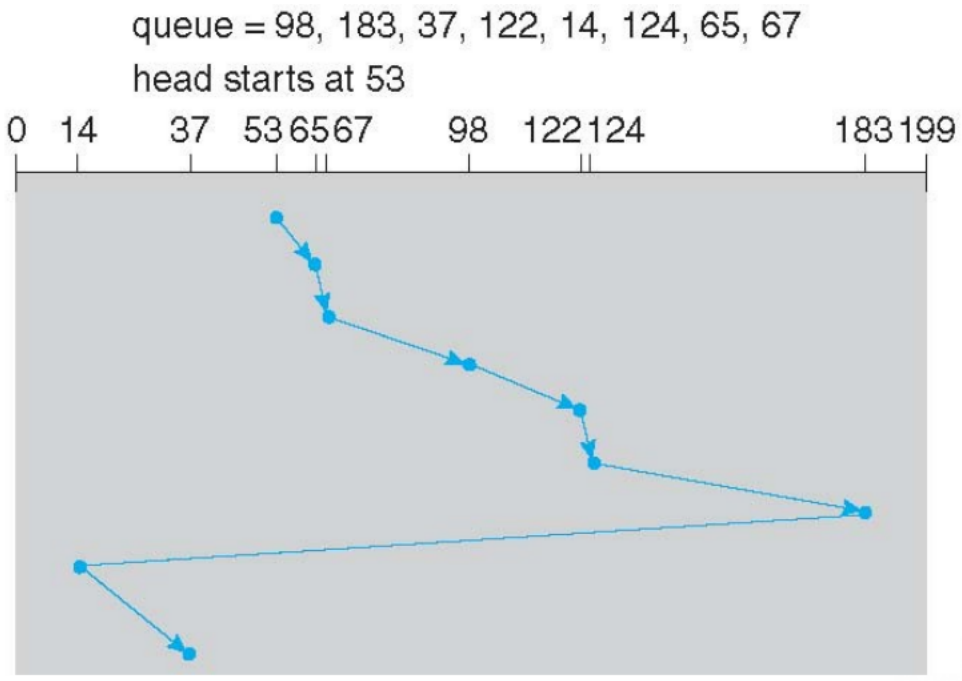
12 C-SCAN

- Provides a more uniform wait time than SCAN
- The head moves from one end of the disk to the other, servicing requests as it goes
  - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one



13 C-LOOK

- LOOK a version of SCAN, C-LOOK a version of C-SCAN
- Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk



14 Selecting a Disk-Scheduling Algorithm

- SSTF is common and has natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk as there is less starvation
- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file-allocation method and metadata layout
- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for the default algorithm
- Rotational latency is difficult for the OS to calculate

15 Disk Management

- Low level formatting, or physical formatting - dividing a disk into sectors that the disk controller can read and write
  - Each sector can hold header information, plus data, plus error correction code
  - Usually 512 bytes of data but can be selectable
- To use a disk to hold files, the operating system still needs to record its own data structures on the disk
  - Partition the disk into one or more groups of cylinders, each treated as a logical disk
  - Logical formatting or ”making a file system”
  - To increase efficiency most file systems group blocks into clusters
    - Disk I/O done in blocks
    - File I/O done in clusters

- Raw disk access for apps that want to do their own block management, keep OS out of the way (databases for example)
- Boot block initialises system
  - The bootstrap is stored in ROM
  - Bootstrap loader program stored in boot blocks of boot partition
- Methods such as sector sparing used to handle bad blocks

16 Swap-Space Management

- Swap-space - Virtual memory uses disk space as an extension of main memory, this is less common now due to memory capacity increases
- Swap-space can be carved out of the normal file system, or, more commonly, it can be in a separate disk partition (raw)
- Swap-Space management
  - 4.3BSD allocates swap space when process starts; holds text segment (the program) and data segment
  - Kernel uses swap maps to track swap space use
  - Solaris 2 allocates swap space only when a dirty page is forced out of physical memory, not when the virtual memory page is first creates
    - File data written to swap space until write to file system requested
    - Other dirty pages go to swap space due to no other home
    - Text segment pages thrown out and reread from the file system as needed
- What if a system runs out of swap space?
- Some systems allow multiple swap spaces

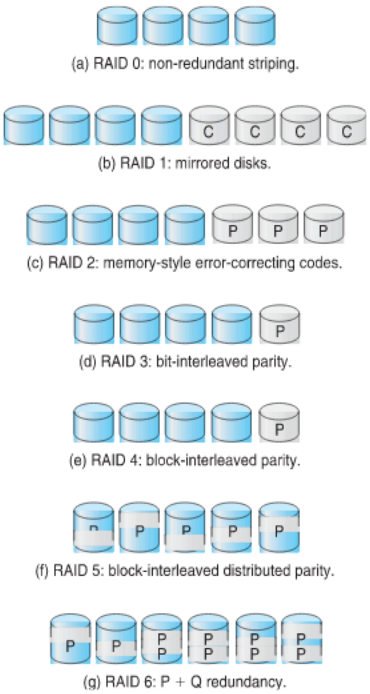
17 RAID Structure

- RAID - Redundant array of inexpensive disks
  - Multiple disk drives provides reliability by redundancy
- Increases mean time to failure
- Mean time to repair - exposure time when another failure could cause data loss
- If mirrored disks fail independently, consider disk with 1300,000 mean time to failure and a 10 hour mean time to repair

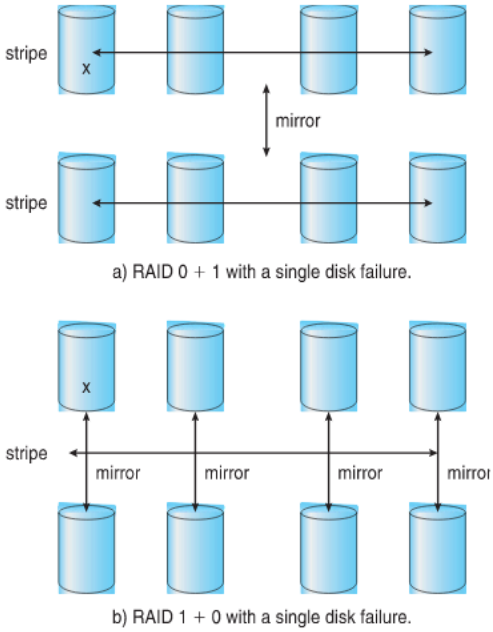
$$\text{Mean time to data loss} 100,000^2/(2 \times 10) = 500 \times 10^6 \text{hours}$$

- Frequently combined with NVRAM to improve write performance
- Several improvements in disk use techniques involve the use of multiple disks working cooperatively
- Disk striping uses a group of disks as one storage unit
- RAID is arranged into six different levels
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
  - Mirroring or shadowing (RAID 1) keeps a duplicate of each disk
  - Stripped mirrors (RAID 1+0) or mirrored stripes (RAID 0+1) provides high performance and high reliability
  - Block interleaved parity (RAID 4,5,6) uses much less redundancy
- RAID within a storage array can still fail if the array fails, so automatic replication of the data between the arrays is common
- Frequently, a small number of hot-spare disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them

18 RAID Levels



19 RAID (0+1) and (1+0)



20 Other Features

- Regardless of where RAID implemented, other useful features can be added
- Snapshot is a view of the file system before a set of changes take place (i.e. at a point in time)
- Replication is automatic duplication of writes between separate sites
  - For redundancy and disaster recovery
  - Can be synchronous or asynchronous

- Hot spare disk is unused, automatically used by RAID production is a disk fails to replace the failed disk and rebuild the RAID set if possible, decreasing mean time to repair

21 Stable - Storage Implementation

- Write-ahead log scheme requires stable storage
- Stable storage means data is never lost (due to failure,etc)
- To implement stable storage:
  - Replicate information on more than one nonvolatile storage media with independent failure models
  - Update information in a controlled manner to ensure that we can recover the stable data after any failure during data transfer or recovery
- Disk write has 1 of 3 outcomes
  1. Successful completion - The data were written correctly on disk
  2. Partial failure - A failure occurred in the midst of a transfer, so only some of the sectors were written with the new data, and the sector being written during the failure may have been corrupted
  3. Total Failure - The failure occurred before the disk write started, so the previous data values on the disk remain intact
- If failure occurs during block write, recovery procedure restores block to a consistent state
  - System maintains 2 physical blocks per logical block and does the following:
    1. Write to 1st physical
    2. When successful, write to 2nd physical
    3. Declare complete only after second write completes successfully

    Systems frequently use NVRAM and one physical to accelerate