Chapter 9: Mass-Storage Structure

Structure of secondary storage devices

Performance characteristics of mass-storage devices

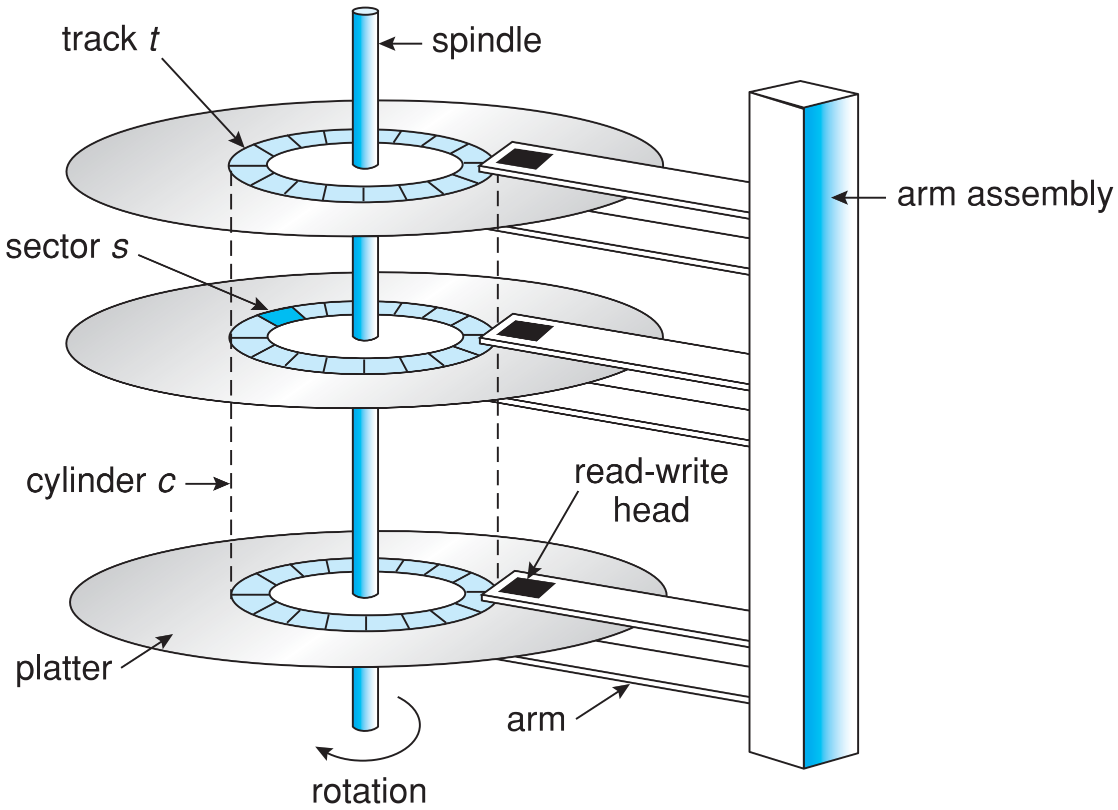
Disk scheduling algorithms

Contents

9.1 Overview of Mass Storage Structure

- Magnetic disks provide bulk of secondary storage of modern com puters
  - Drives rotate at 60 to 250 times per second
  - Transfer rate is rate at which data flow between drive and 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 he ad (rotational latency)
  - Head crash results from disk head making contact with the disk surface
- Disks can be removable
- Drive attached to computer via I/O bus
  - Busses vary, including EIDE, ATA, SATA, USB, Fibre Channel, SCSI, SAS, Firewire
  - Host controller in computer uses bus to talk to disk controller built into dri ve or storage array

Moving-head Disk Mechanism



Hard Disks

- Platters range from .85” to 14” (historically)
  - Commonly 3.5”, 2.5”, and 1.8”
- Range from 30GB to 3TB per drive
- Performance
  - Transfer Rate – theoretical – 6 Gb/sec
  - Effective Transfer Rate – real – 1Gb/sec
  - Seek time from 3ms to 12ms – 9ms common for desktop drives
  - Average seek time measured or calculated based on 1/3 of tracks
  - Latency based on spindle speed
    - 1 / (RPM / 60) = 60 / RPM
  - Average latency = ½ latency

Hard Disk Performance

- Access Latency = Average access time = average seek time + average latency
  - For fastest disk 3ms + 2ms = 5ms
  - For slow disk 9ms + 5.56ms = 14.56ms
- Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead
- For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead =
  - 5ms + 4.17ms + 0.1ms + transfer time =
  - Transfer time = 4KB / 1Gb/s * 8Gb / GB * 1GB / 1024^2KB = 32 / (1024^2) = 0.031 ms
  - Average I/O time for 4KB block = 9.27ms + .031ms = 9.301ms

Disk Structure

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

Solid-State Disks

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

Magnetic Tape

- Was early secondary-storage medium
  - Evolved from open spools to cartridges
- 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
- Once data under head, transfer rates comparable to disk
  - 140MB/sec and greater
- 200GB to 1.5TB typical storage

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

Disk Scheduling (Cont.)

- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- I/O request includes input or output mode, disk address, me mory address, number of sectors to transfer
- OS maintains queue of requests, per disk or device
- Idle disk can immediately work on I/O request, busy disk mea ns work must queue
  - Optimization algorithms only make sense when a queue exists

Disk Scheduling (Cont.)

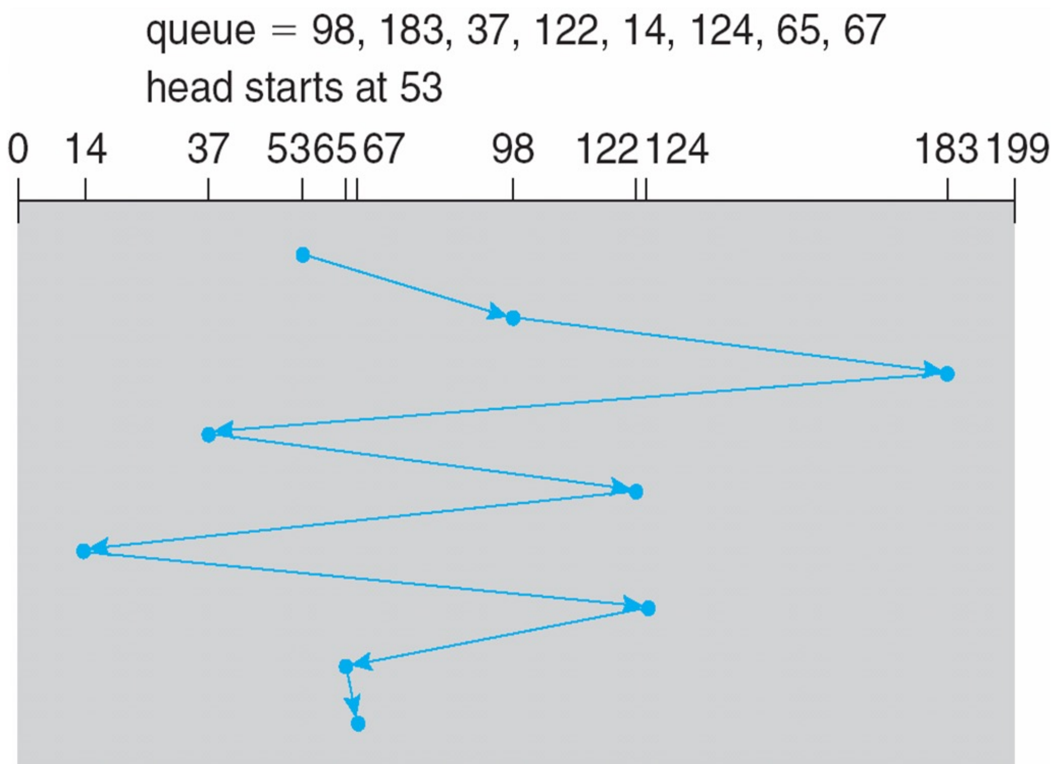
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199)

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

Head pointer 53

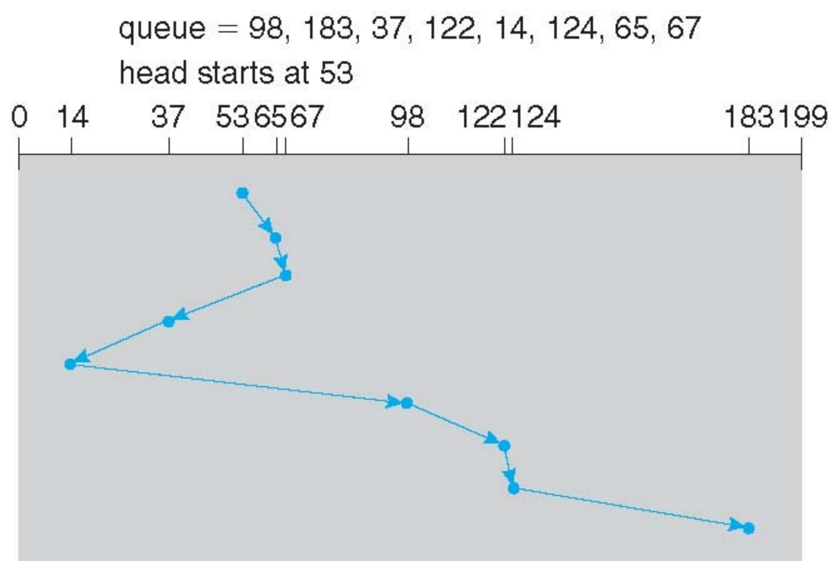
FCFS

Illustration shows total head movement of 640 cylinders



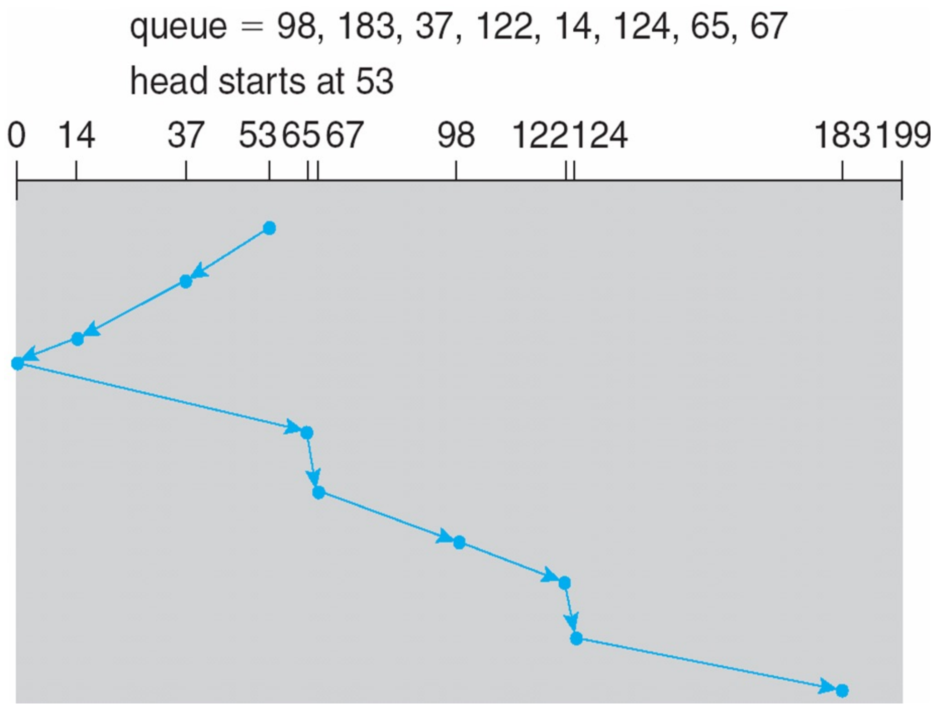
SSTF

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



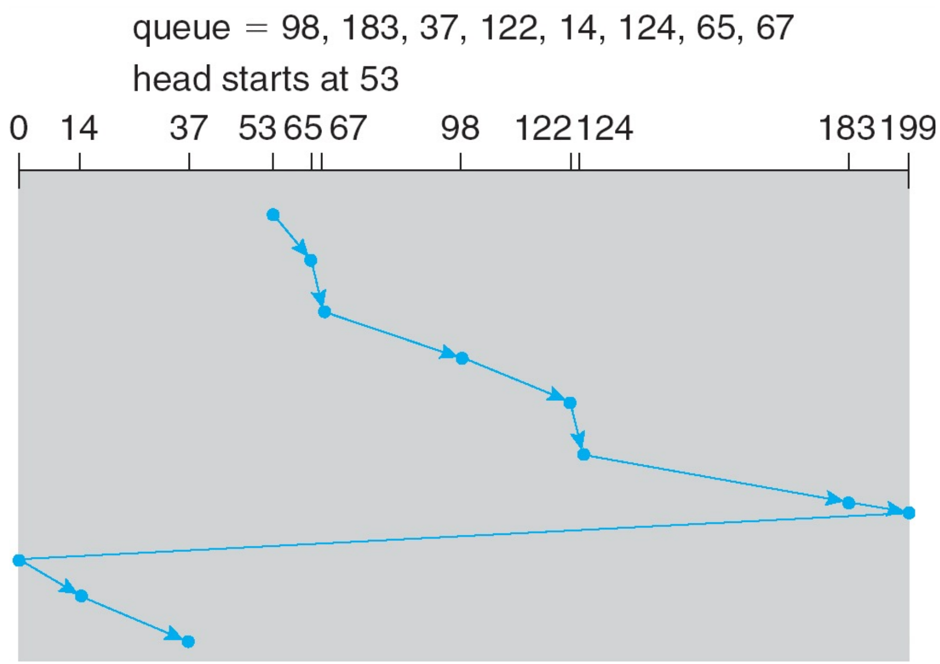
SCAN

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



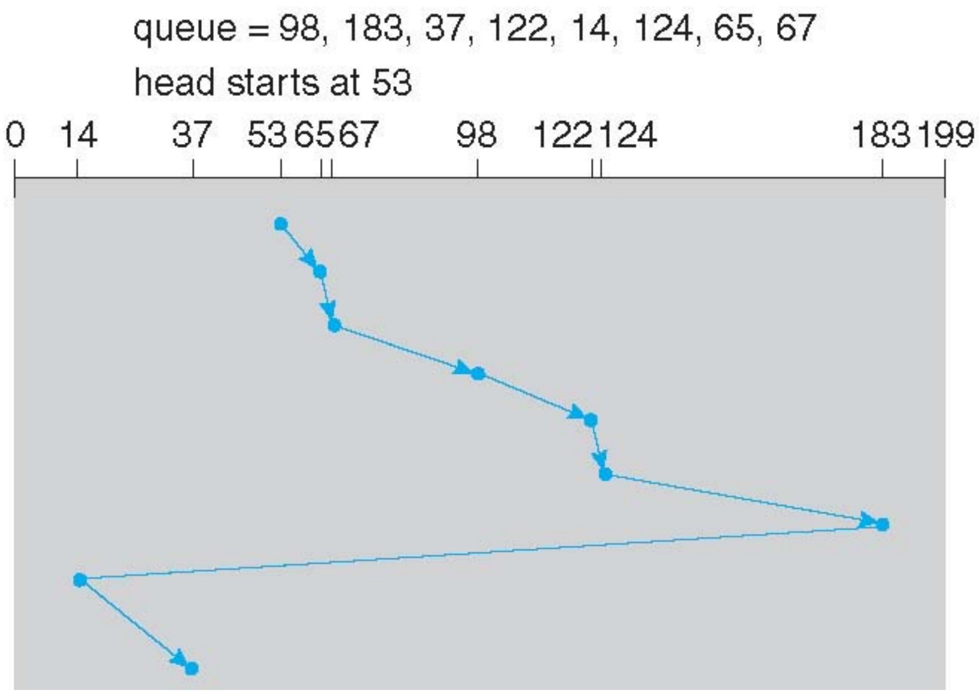
C-SCAN

- Provides a more uniform wait ti me than SCAN
- The head moves from one end of the disk to the other, servici ng requests as it goes
  - When it reaches the other end, h owever, it immediately returns to the beginning of the disk, withou t servicing any requests on the re turn trip
- Treats the cylinders as a circula r list that wraps around from t he last cylinder to the first one
- Total number of cylinders?



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
- Total number of cylinders?



Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
  - 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 operatin g system, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for the default algorithm
- What about rotational latency?
  - Difficult for OS to calculate
- How does disk-based queueing effect OS queue ordering efforts?

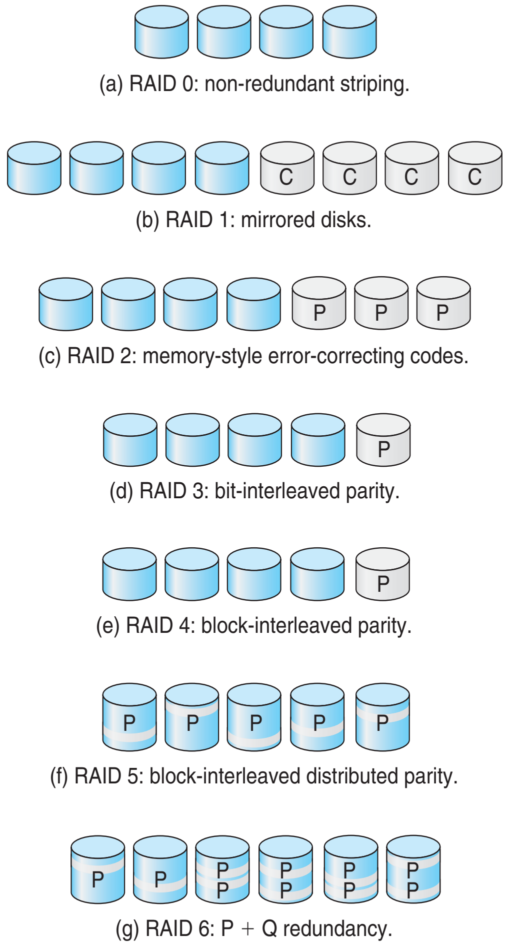
9.7 RAID Structure

- RAID – redundant array of inexpensive (independent) disks
  - Improvement of reliability via redundancy
  - Improvement in performance via parallelism
- Mirroring - improves the reliability of the storage system by storing red undant data
- Mean time to failure of a mirrored disk can be affected by two factors: single disk's mean time to failure and mean time to repair
  - If mirrored disks fail independently, consider disk with 100,000hr mean time to fail ure and 10 hour mean time to repair
  - Mean time to data loss is $100,000^2 / (2 * 10) = 500 * 106$ hours, or 57,000 years!
- Data striping
  - Bit-level striping
  - Block-level striping

RAID (Cont.)

- RAID is arranged into six different levels
- RAID schemes improve performance and improve the reliabili ty of the storage system by storing redundant data
  - Mirroring or shadowing (RAID 1) keeps duplicate of each disk
  - Block interleaved parity (RAID 4, 5, 6) uses much less redundancy
  - Striped mirrors (RAID 1+0) or mirrored stripes (RAID 0+1) provides h igh performance and high reliability
- Frequently, a small number of hot-spare disks are left unalloc ated, automatically replacing a failed disk and having data re built onto them

RAID Levels



(a) RAID 0: non-redundant striping.
C C C C
(b) RAID 1: mirrored disks.
P P P
(c) RAID 2: memory-style error-correcting codes.
P
(d) RAID 3: bit-interleaved parity.
P
(e) RAID 4: block-interleaved parity.
P P P P P
(f) RAID 5: block-interleaved distributed parity.
P P P P P P P P P P
(g) RAID 6: P + Q redundancy.

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