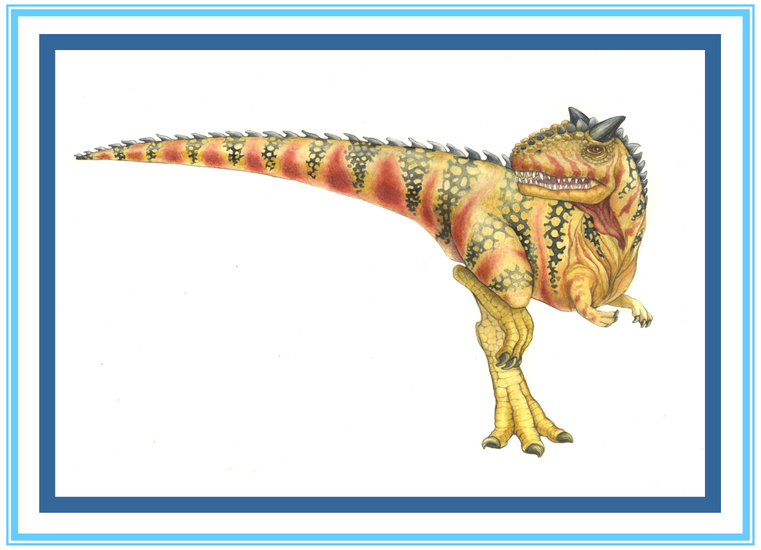
Chapter 11: Mass-Storage Systems



Chapter 11: Mass-Storage Systems

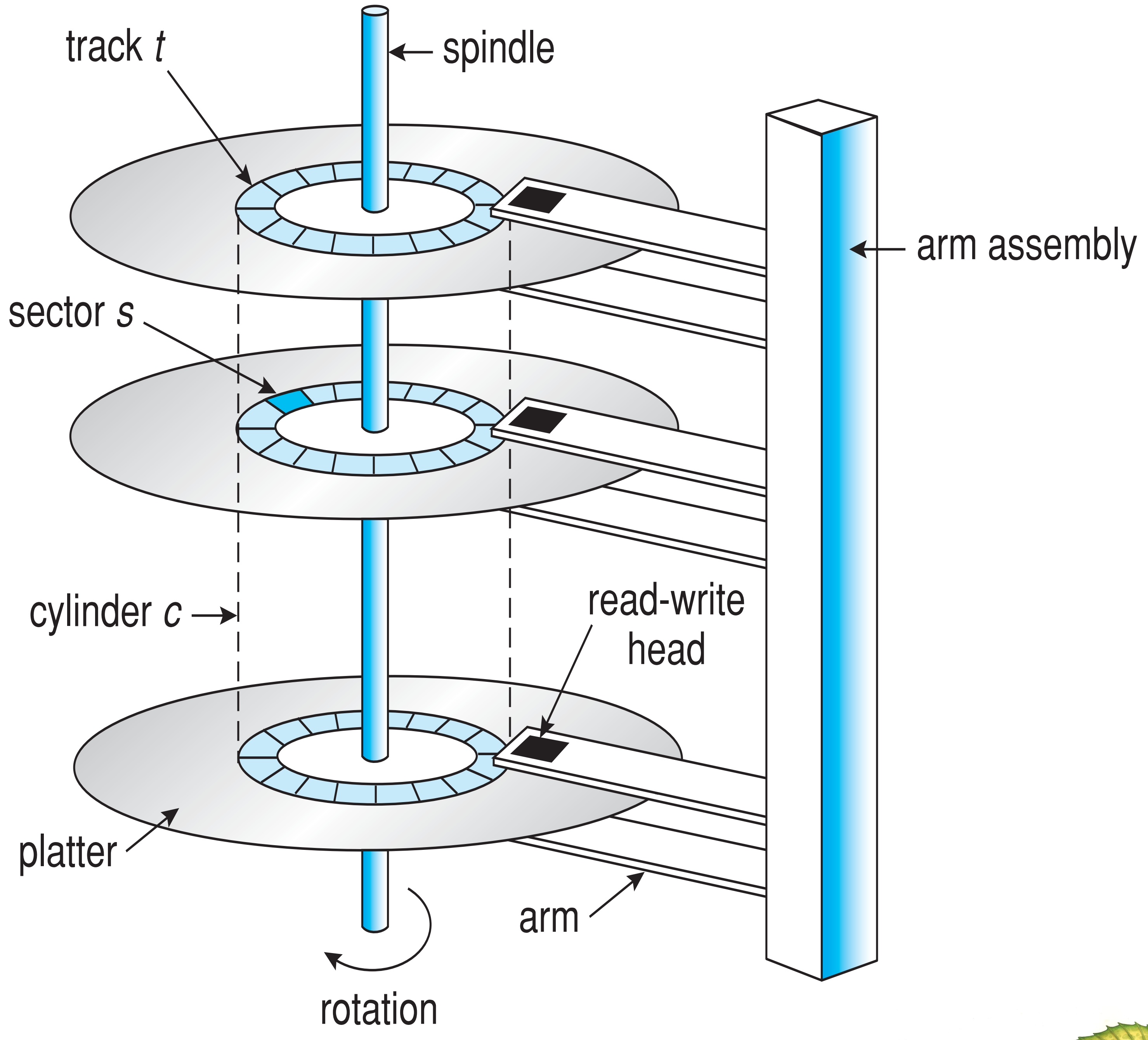
- Overview of Mass Storage Structure
- Disk Structure
- Disk Scheduling
- RAID Structure

Objectives

- Describe the physical structure of secondary storage devices and the effect of a device's structure on its uses
- Explain the performance characteristics of mass-storage devices
- Evaluate I/O scheduling algorithms
- Discuss operating-system services provided for mass storage, including RAID

Moving-head Disk Mechanism

- Each disk platter has a flat circular shape with diameters of 1.8, 2.5 to 3.5 inches
- Two surfaces of a platter covered with a magnetic materials for storing information
- A read-write head “flies” just above each surface of every platter
- The heads are attached to a disk arm that move all heads as a unit
- The surface of a platter is logically divided into circular tracks, which are subdivided into hundreds of sectors (per track)
- The set of tracks that are at one arm position makes up a cylinder - thousands of concentric cylinders in a disk drive
- There could be thousands of concentric cylinders in a disk drive

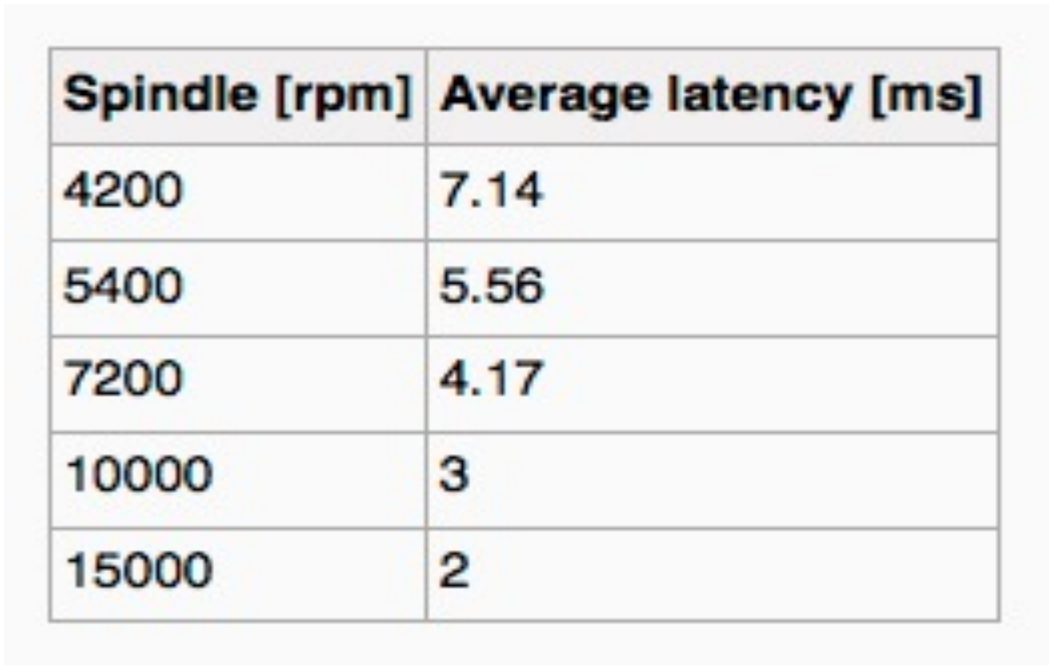


Overview of Mass Storage Structure

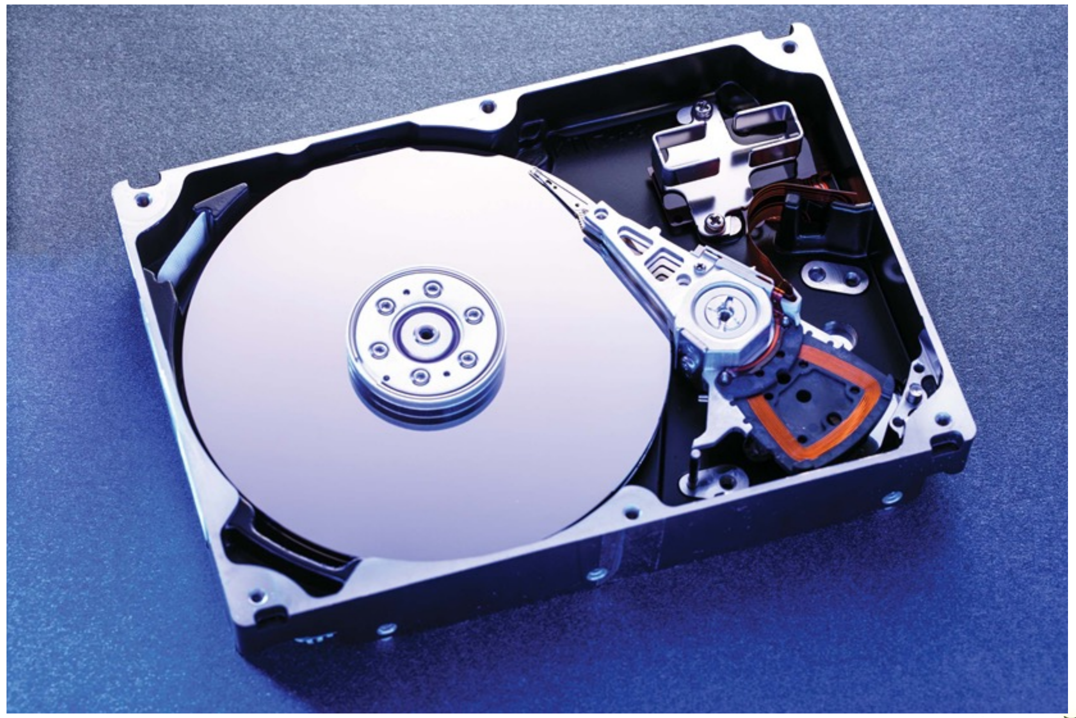
- **Magnetic disks** provide bulk of secondary storage of modern computers
  - Drives rotate at 60 to 250 times per second
  - Transfer rate is rate at which data flow between drive and computer
  - Positioning time (random-access time) is time to (1) move disk arm to desired cylinder (seek time) and (2) time for desired sector to rotate under the disk head (rotational latency)
  - Head crash results from disk head making contact with the disk surface -- that's bad!
- 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 drive or storage array

Hard Disk Drive

- 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
  - RPM typically, 5,400, 7,200, 10,000 and 15,000
  - Latency based on spindle speed
    - 1/(RPM/60) = 60/RPM
  - Average latency = ½ latency
  - For example with 7200 rpm, that is 120 rps, the average latency is 1/240 = 4.17 mini-seconds



Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2



Hard Disk Performance

- Access latency or average access time = average seek time + average latency
  - For fast 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/s transfer rate with a 0.1ms controller overhead =
  - 5ms + 4.17ms + 4KB / 1Gb/sec + 0.1ms =
  - 9.27ms + 4 / 131072 sec =
  - 9.27ms + .12ms = 9.39ms
  - Thus, it takes an average 9.39ms to transfer 4KB, thus effective bandwidth is 4KB/9.39ms = 3.408 Mb/sec only (with a transfer rate at 1 Gb/sec given the overhead).

Hard Disk Performance (Cont.)

- There is a huge gap in hard drive performance between random and sequential workloads
- Consider a disk with 300GB capacity, average seek time is 4 milliseconds (4 ms), RPM is 15,000 RPM or 250 RPS (the average rotation time is 2 ms), transfer rate is 125MB/s, and a 4KB read occurs at a random location, recall

  average access time = average seek time + average latency

- Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead (ignored) = 4 ms + 2 ms + 30 microseconds
- The effective bandwidth or transfer rate is 0.66MB/s

- With sequential access of a 100 MB file, suppose there is only one seek and rotation, this would yield an effective bandwidth or transfer rate close to 125MB/s

Solid-State Disk (SSD)

- An SSD is nonvolatile memory (NVM) used like a hard drive
  - Many technology variations, e.g., from DRAM with a battery to maintain its state in a power failure, through flash-memory technologies like single-level cell (SLC) and multilevel cell (MLC) chips
  - SSDs can be more reliable than HDDs because they have no moving parts
  - They are much faster because they have no seek time or rotation latency.
  - They consumes less power – power efficiency
  - But they are more expensive per MB, have less capacity, and may have shorter life span
- Because they are much faster than magnetic disk drives, standard bus interface can be too slow, causing a major limit on throughput
  - Some connect directly to system bus (e.g., PCI)
  - Some use them as a new cache tier, moving data between magnetic disk, SSDs, and memory to optimize performance

Magnetic Tape

- Magnetic tape was an early secondary-storage medium
  - It is relatively permanent and can hold large quantities of data
  - Access time is slow, as moving to the correct spot on a tape can take minutes
  - Random access ~1000 times slower than magnetic disk, so not very useful for secondary storage
- Mainly used for backup, storage of infrequently-used data, or as a medium of transferring information from one system to another
- Tape capacities vary greatly, depending on the particular kind of tape drive, with current capacities exceeding several terabytes, and typically between 200GB and 1.5TB

Disk Structure

- Disk drives are addressed as large one-dimensional arrays of logical blocks, where a logical block is the smallest unit of transfer。In another word, disk is represented by a number of disk blocks, each block has a unique block number.
  - The size of a logical block is usually 512 bytes
  - Low-level formatting creates logical blocks on physical media

- The one-dimensional array of logical blocks is mapped into sectors 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 in that cylinder, and then through the rest of the cylinders from outermost to innermost
  - Logical to physical address (consist of a *cylinder number*, a *track number* within the cylinder, and a *sector number* with the track) should be easy, except
    - Defective sectors, mapping hides this by substituting spare sectors from elsewhere on disk
    - The number of sectors per track may not be a constant on some devices. Non-constant # of sectors per track via constant angular velocity

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having fast access time and large disk bandwidth

- The seek time is the time for the disk head arm to move the heads to the corresponding cylinder containing the desired sector, which can be measured by the seek distance in term of number cylinders/tracks.

- The rotational latency is the additional time for the disk to rotate the desired sector to the disk head.

- 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.

- We can improve both access time and the bandwidth by managing the order in which disk I/O requests are serviced.

Disk Scheduling (Cont.)

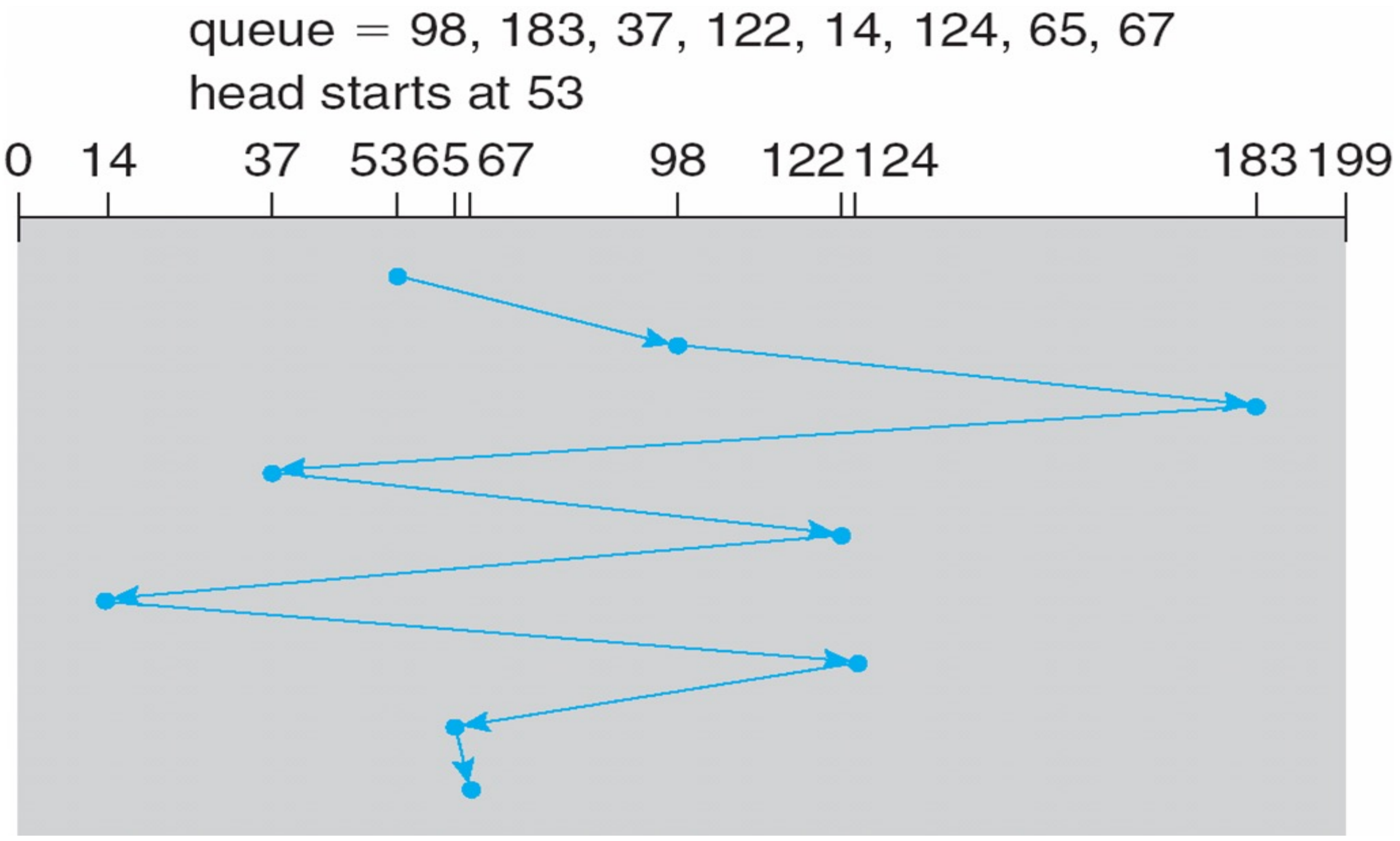
- There are many sources of disk I/O requests, from OS, system processes, and user processes
  - I/O request includes input/output modes, disk address, memory address, number of sectors to transfer
- OS maintains a queue of requests per disk or device
  - For a multiprogramming system with many processes, the disk queue often has several pending requests.
- Idle disk can immediately work on I/O request, busy disk means work must be queued.
  - Optimization only make sense when a queue exists.
- Disk drive controllers have small buffers and manage a queue of I/O requests (of varying “depth”).
  - When one request is completed, which pending request to select to service next?
  - **Disk scheduling**
- We next illustrate scheduling algorithms with a request queue (0-199), 0-199 are cylinder numbers.

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

The current Head position is 53

FCFS First-Come-First-Serve

- FCFS is intrinsically fair, but it generally does not provide the fastest service
- Illustration shows total head movement of **640** cylinders

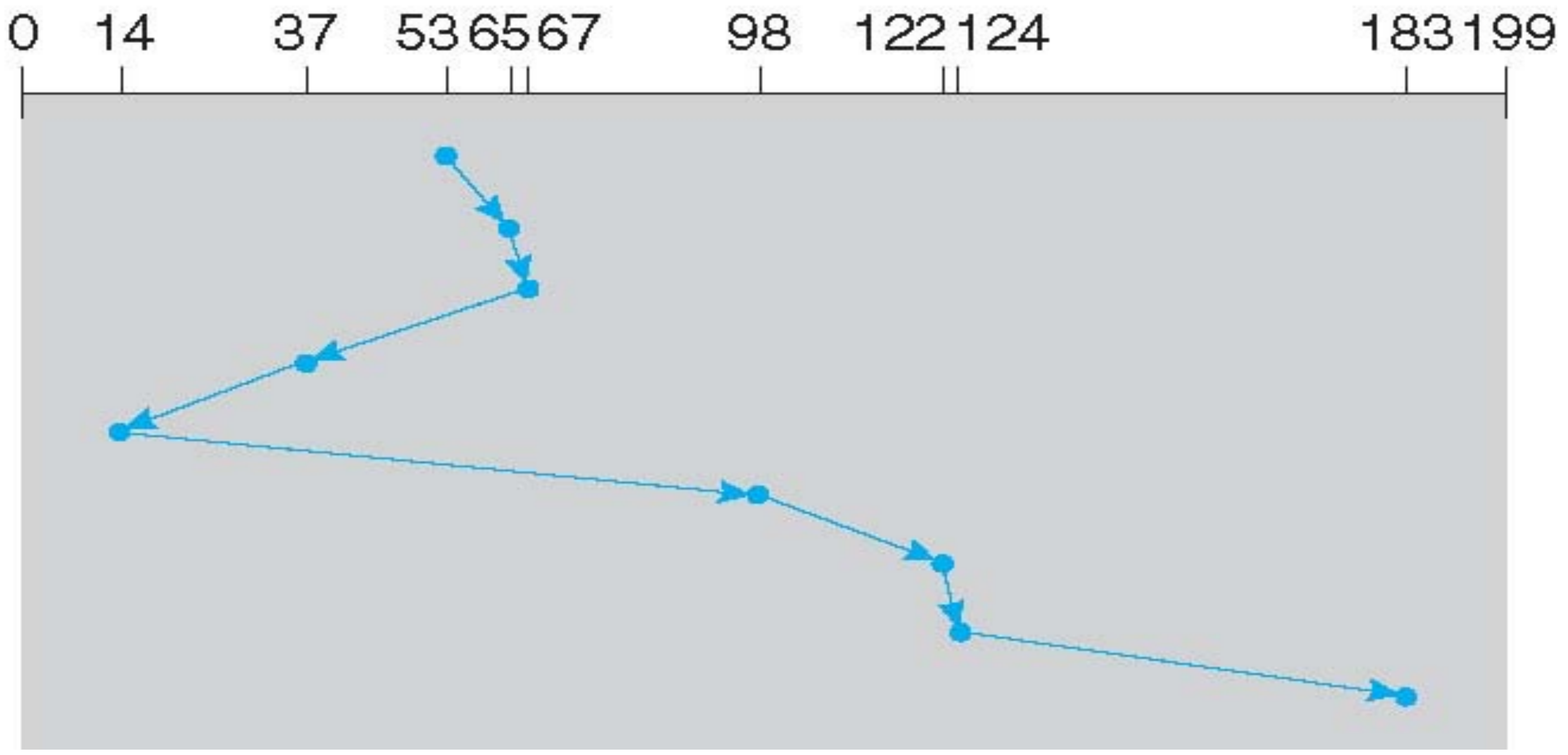


SSTF Shortest-Seek-Time-First

- The Shortest Seek Time First (SSTF) selects the request with the least seek time from the current head position, i.e., choose the pending request closest to the current head position (either direction)
- SSTF scheduling is a form of SJF scheduling (greedy algorithm); may cause starvation of some requests, as requests may arrive at any time dynamically
- Illustration shows total head movement of **236** cylinders

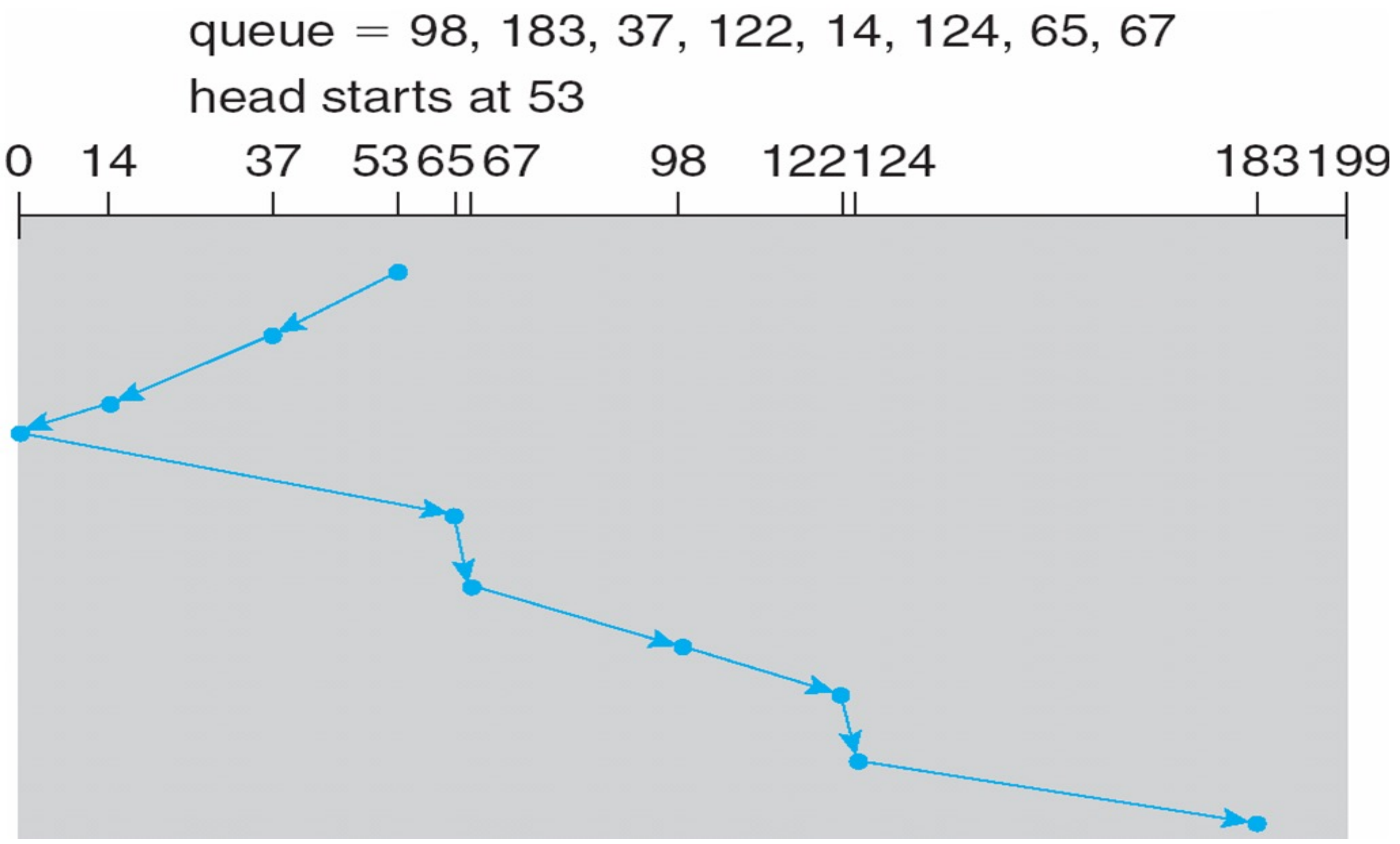
queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



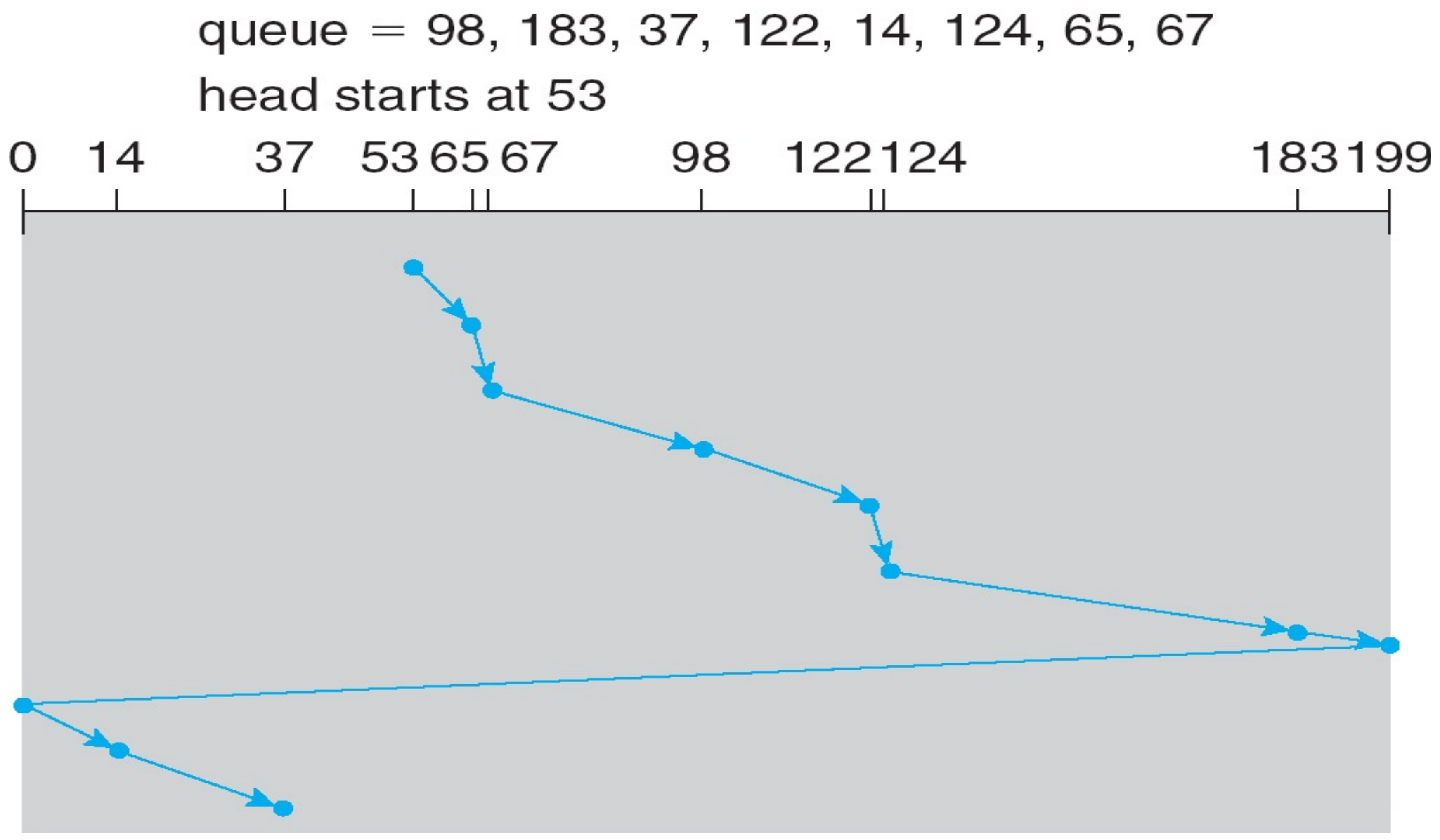
SCAN Scheduling

- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests as it reaches each cylinder, until it gets to the other end of the disk. At the other end, the direction of head movement is reversed, and servicing continues. This sometimes called the elevator algorithm.
- Note if requests are uniformly distributed across cylinders, the heaviest density of requests are at other end of disk and those wait the longest. Also we need to know direction of head movement.
- Illustration shows total head movement of **236** cylinders



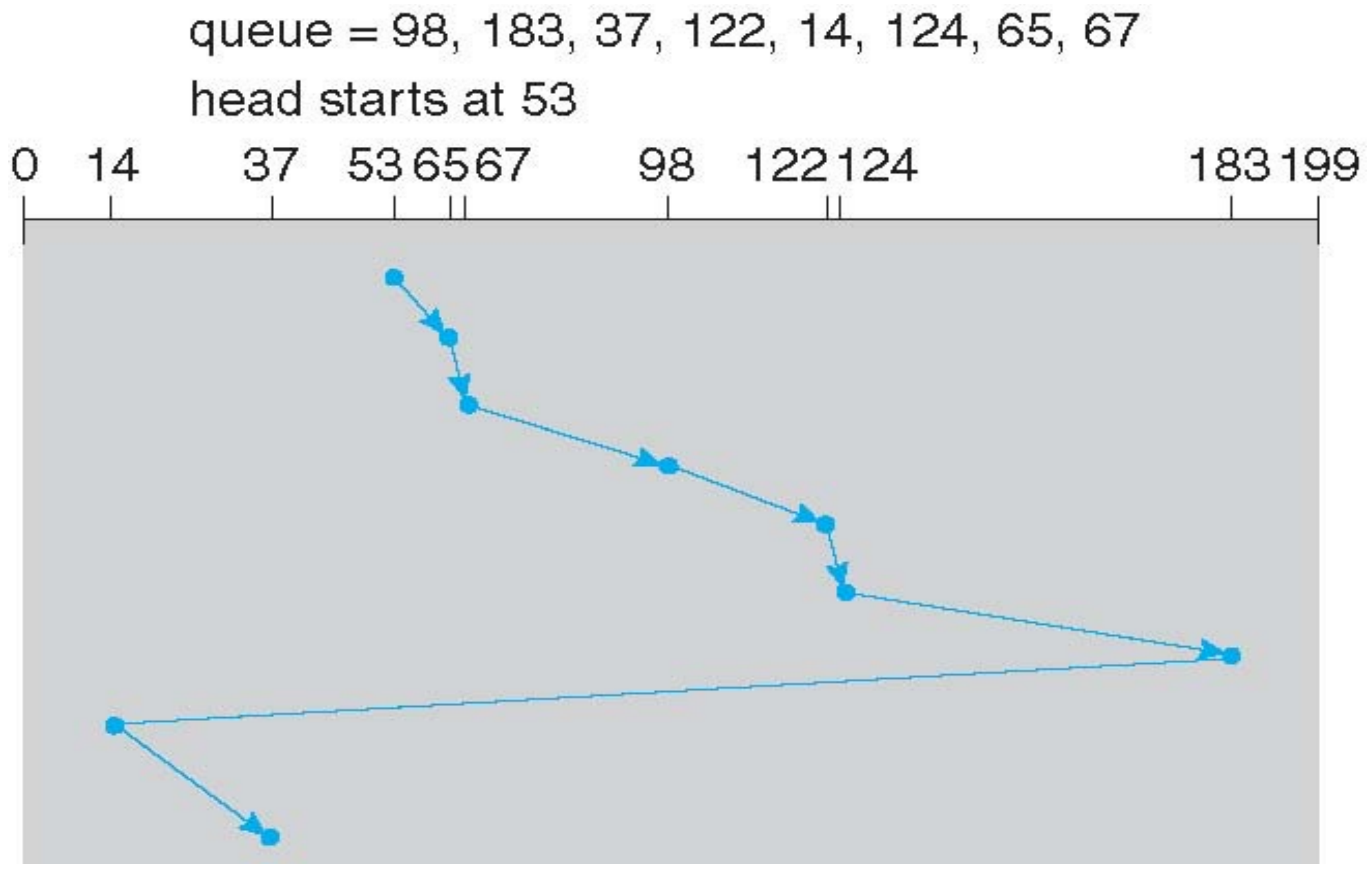
C-SCAN

- **C-SCAN**, Circular-SCAN, a variant of SCAN, provides a more uniform waiting 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
- Total number of cylinders - **382**



C-LOOK

- LOOK a version of SCAN, C-LOOK a version of C-SCAN
- Disk 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? – **322** (for C-LOOK) and **308** for LOOK



Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal as it increases performance over FCFS. SCAN and C-SCAN perform better for systems that place a heavy load on the disk, because they are less likely to cause starvation problems.
- The scheduling performance also depends on the number and type of requests. If only one request, all scheduling algorithms behave the same (like FCFS scheduling)
- Requests for disk service can be greatly influenced by the file-allocation method (to be discussed)
  - Contiguously allocated file will generate several requests close together on the disk, resulting in limited head movement, while a linked or indexed file may include blocks widely scattered on the disk, resulting in greater head movement.
- The location of directories and index blocks are also important, which are accessed frequently.
  - Directory entry and file data on different cylinders cause excessive head movement
  - Caching directory and index block in memory help
- 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 as the default algorithm.

RAID - Improving Reliability via Redundancy

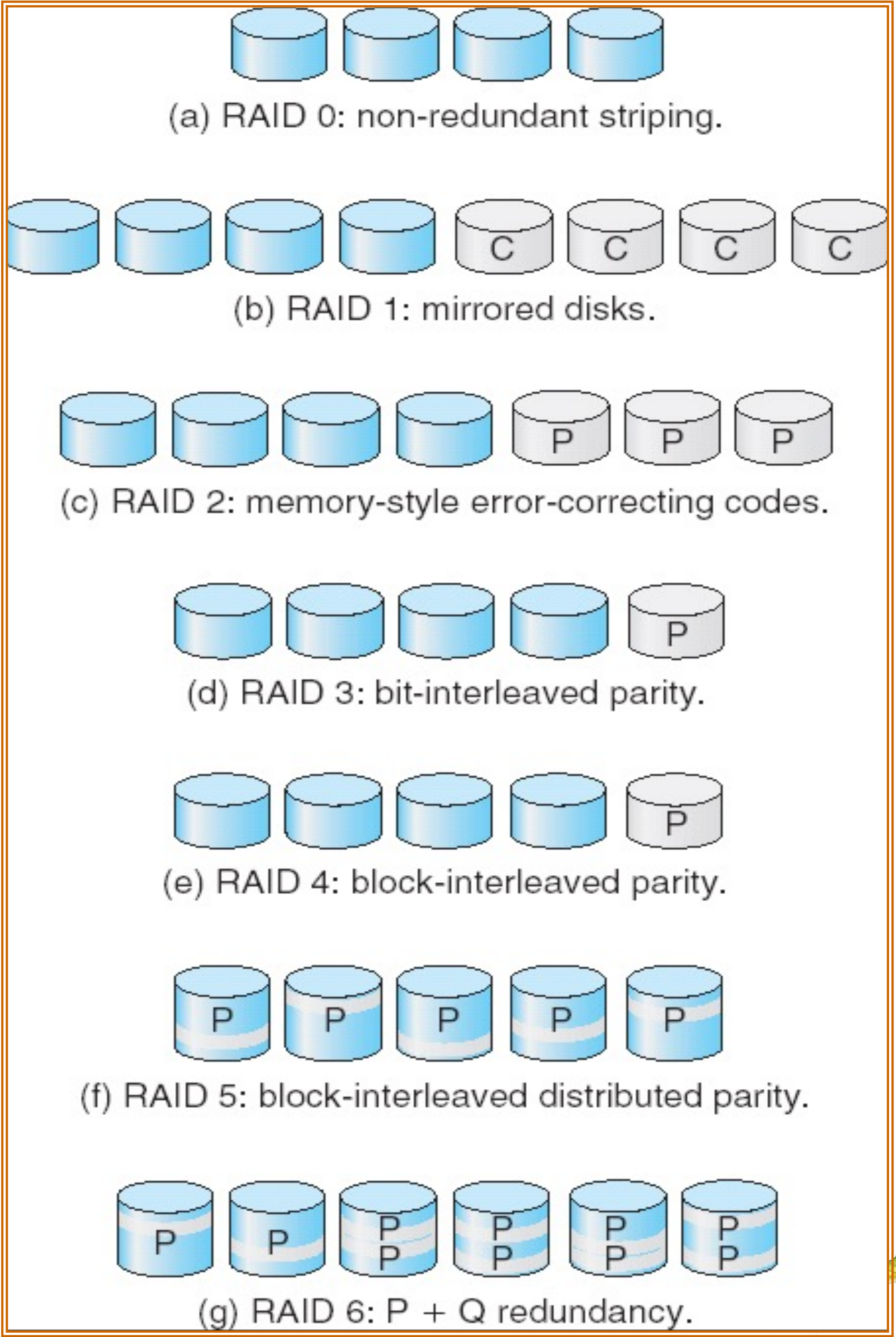
- RAID – Redundant Arrays of Independent Disks
  - In the past, RAID composed of small, cheap disks were viewed as a cost-effective alternative to large, expensive disks (once called redundant arrays of inexpensive disks)
  - Now RAIDs are used for higher reliability via redundancy and higher data-transfer rate (access in parallel)
- Increases mean time to failure
  - The chance that a disk out of N disks fails is much higher than the chance that a specific single disk fails. Suppose that the mean time to failure of a single disk is 100,000 hours, the mean time to failure of some disk in an array of 100 disks will be 100,000/100 = 1,000 hours, or 41.66 days!
- The data loss rate is unacceptable if we store only one copy of the data
  - The solution is to introduce redundancy; the simplest (but most expensive) approach is to duplicate every disk, called mirroring. Every write is carried out on two physical disks. Data will be lost only if the second disk fails before the first failed disk is replaced.
- The mean time to repair is the time it takes (on average) to replace a failed disk and to restore data on it – exposure time when another failure could cause data loss
  - Suppose the mean time to failure of a single disk is 100,000 hours and the mean time to repair is 10 hours. The mean time to data loss is $100,000^2 / (2 * 10) = 500 * 10^6$ hours, or 57,000 years!

RAID – Improving Performance via Parallelism

- Parallelism in a disk system, via data striping, has two main goals:
  - Increase the throughout of multiple small access by load balancing
  - Reduce the response time of large access
- Bit-level striping
  - For example, if we have an array of 8 disks, we can write bit *i* of each byte to disk *i*. The array of 8 disks can be treated as a single disk with sectors that are 8 times the normal sector size. The access rate can be improved by 8 times!
  - Bit-level striping can be generalized to include a number of disks that either is a multiple of 8 or divides 8. For example, with an array of 4 disks, bit *i* and *4+i* of each byte cane be stored in disk *i*
- Block-level striping
  - Blocks of a file are striped across multiple disks
  - With *n* disks, block *i* of a file goes to disk (*i* mod *n*)+*i*

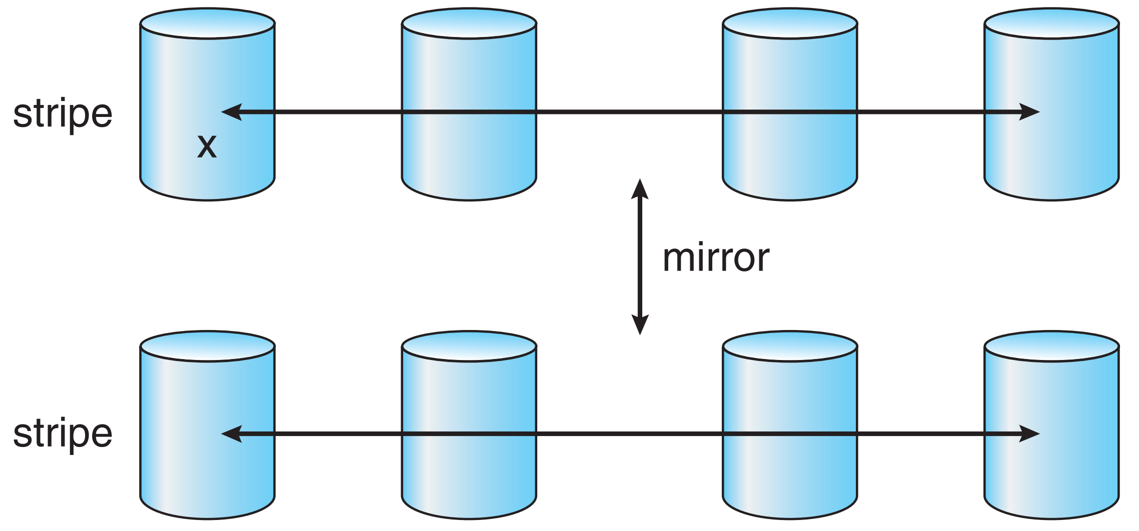
RAID Structure

- Mirroring provides high reliability, but expensive. Striping provides high data-transfer rates, but does not provide reliability
- Many schemes provide redundancy at lower cost by using striping combined with “parity” bits, generally, classified into 6 RAID levels
- In the RAID levels, 4 disks’ worth of data are stored. P indicates error-correcting bits, C indicates a second copy of the data
  - RAID 0 refers to disk array with striping at the level of blocks with non redundancy
  - Mirroring or shadowing (RAID 1) keeps duplicate of each disk
  - Striped 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

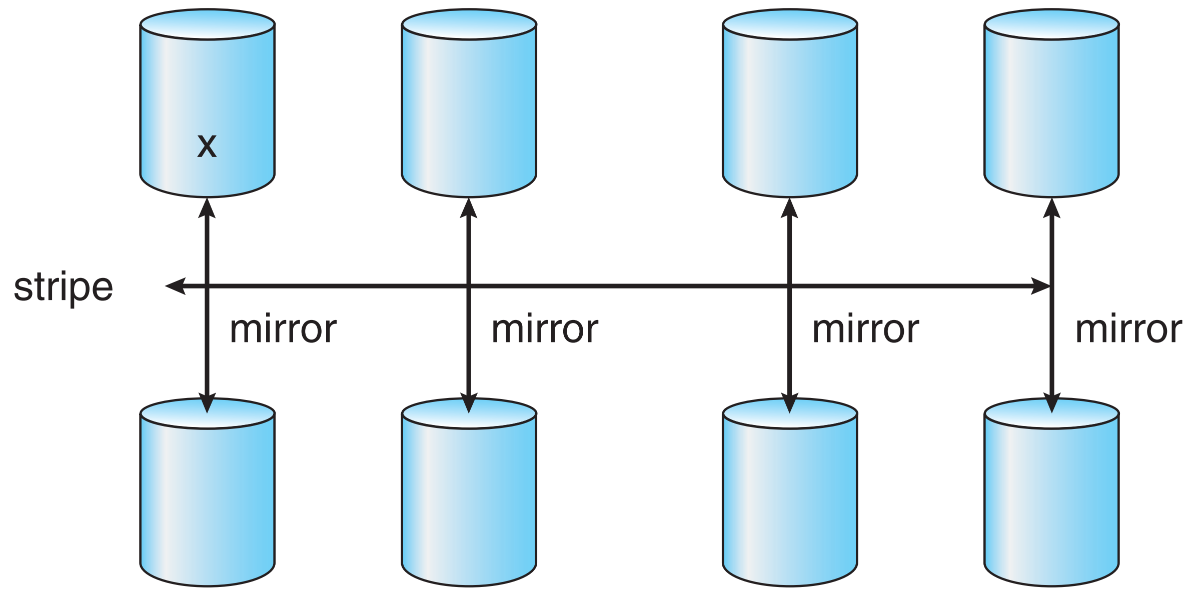

(a) RAID 0: non-redundant striping.
C C C C
(b) RAID 1: mirrored disks.
P P P
(c) RAID 2: memory-style error-correcting codes.
P
(d) RAID 3: bit-interleaved parity.
P
(e) RAID 4: block-interleaved parity.
P P P P P
(f) RAID 5: block-interleaved distributed parity.
P P P P P P P P P P
(g) RAID 6: P +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

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

- Both striped mirrors (RAID 1+0) or mirrored stripes (RAID 0+1) provides high performance (RAID 0) and high reliability (RAID 1)
- (RAID 0+1) a set of drives are striped, and then the stripe is mirrored to another, equivalent stripe
- (RAID 1+0) drives are mirrored in pairs and then the resulting mirrored pairs are striped



a) RAID 0 + 1 with a single disk failure.



b) RAID 1 + 0 with a single disk failure.

Other Features

- Regardless of what RAID implemented, other useful features can be added at each level
- Snapshot is a view of file system before the last update took place (for recovery)
- Replication is automatic duplication of writes between separate sites for redundancy and disaster recovery. This can be synchronous (each block must be written locally and remotely before the write is considered complete) or asynchronous (writes are grouped together and written periodically)
- A hot spare disk is not used for data but is configured to be used as a replacement in case of disk failure
  - For instance, a hot spare can be used to rebuild a mirrored pair should one of the disks in the pair fails. In this way, RAID level can be reestablished automatically, without waiting for the failed disk to be replaced/repaired.

End of Chapter 11