Storage Systems

Jo, Heeseung

Today's Topics

HDDs (Hard Disk Drives)

Disk scheduling policies

Secondary Storage

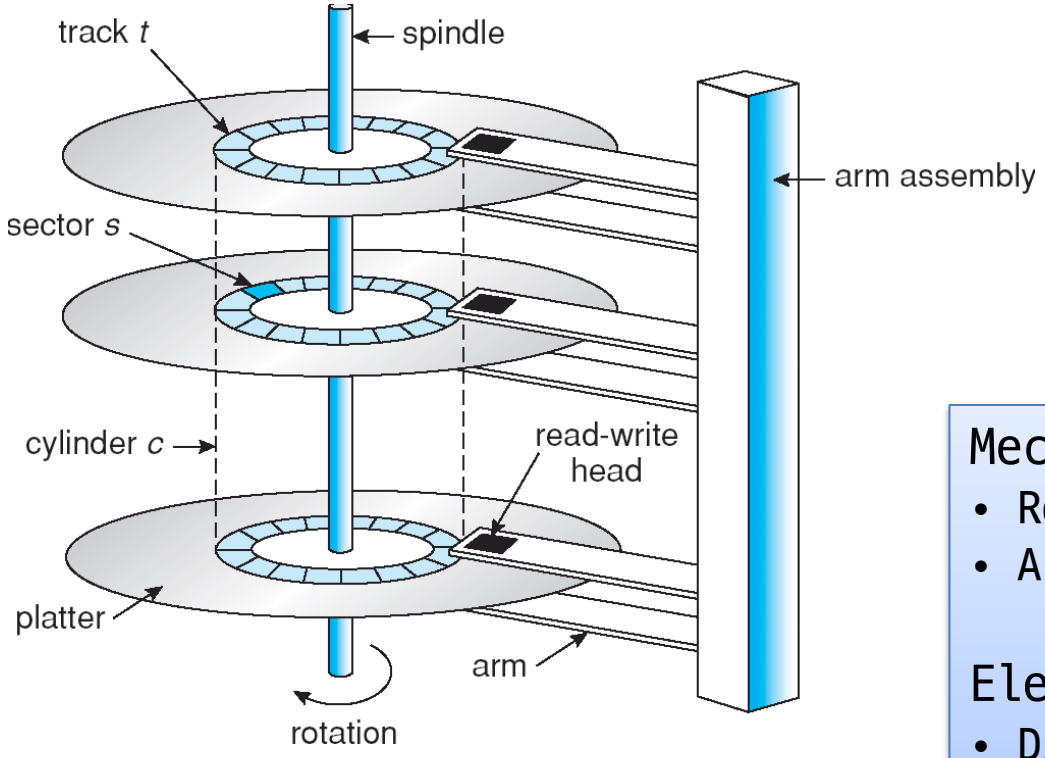
Secondary storage usually

- is anything that is outside of "primary memory"
- does not permit direct execution of instructions
- does not permit direct data read/write

Characteristics

- It's large: 4TB and more
- It's cheap
- It's persistent: data survives power loss
- It's slow: milliseconds to access

HDDs (1)

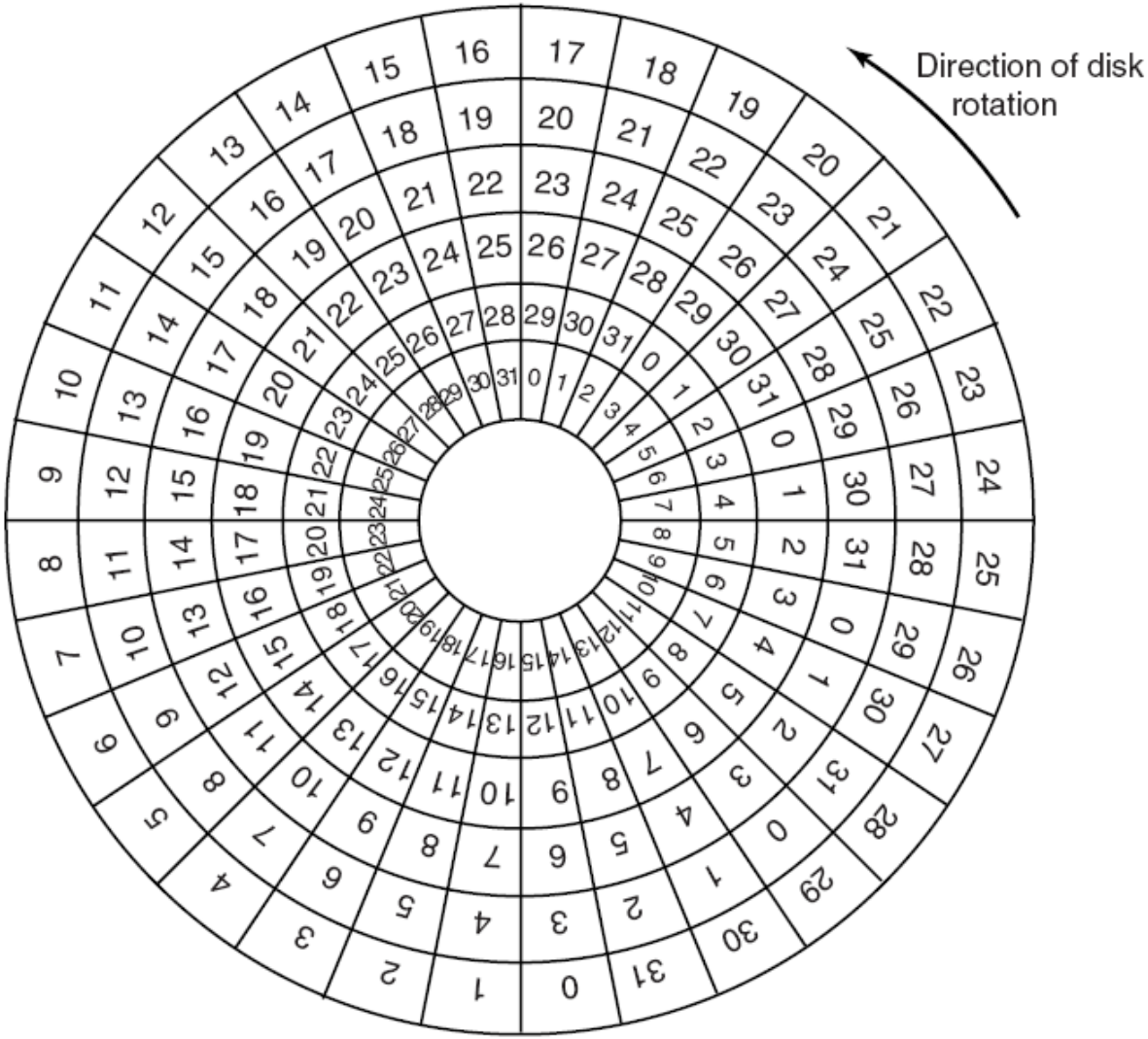


Mechanical
- Rotating disks
- Arm assembly

Electronics
- Disk controller
- Buffer
- Host interface

HDDs (2)

Disk layout



HDDs (3)

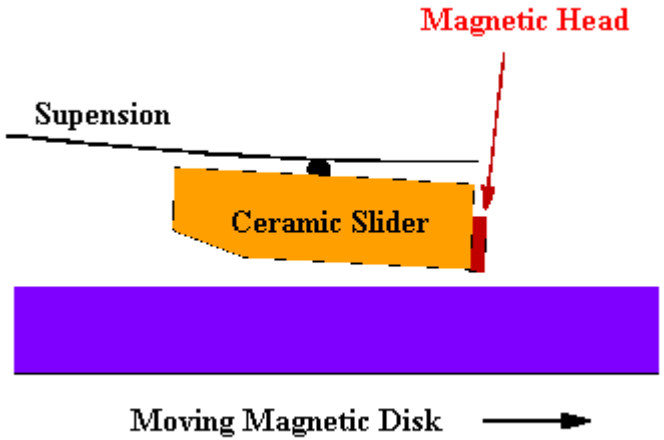
Seagate Barracuda ST31000528AS (1TB)

- 4 Heads, 2 Discs
- Max. recording density: 1413K BPI (bits/inch)
- Avg. track density: 236K TPI (tracks/inch)
- Avg. areal density: 329 Gbits/sq.inch

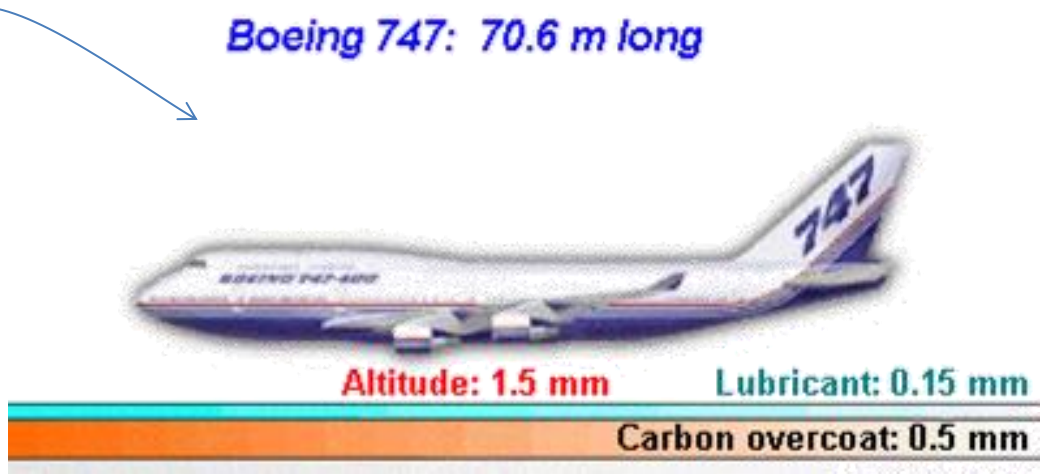
- Spindle speed: 7200rpm (8.3ms/rotation)
- Average seek time: < 8.5ms (read), < 9.5ms (write)
- Max. internal data transfer rate: 1695 Mbits/sec
- Max. I/O data transfer rate: 300MB/sec (SATA-2)
- Max. sustained data transfer rate: 125MB/sec
- Internal cache buffer: 32MB
- Max power-on to ready: < 10.0 sec

HDDs (4)

Hard disk internals



Scale up to airplane



- Our Boeing 747 will fly
  - At the altitude of only a few mm
  - At the speed of approximately 65mph(=100km/h)
  - Periodically landing and taking off
- The surface of the runway
  - consist of a few mm-thick layers
  - will stay intact for years

Managing Disks (1)

Disks are messy physical devices:

- Errors
- Bad blocks
- Missed seeks
- etc.

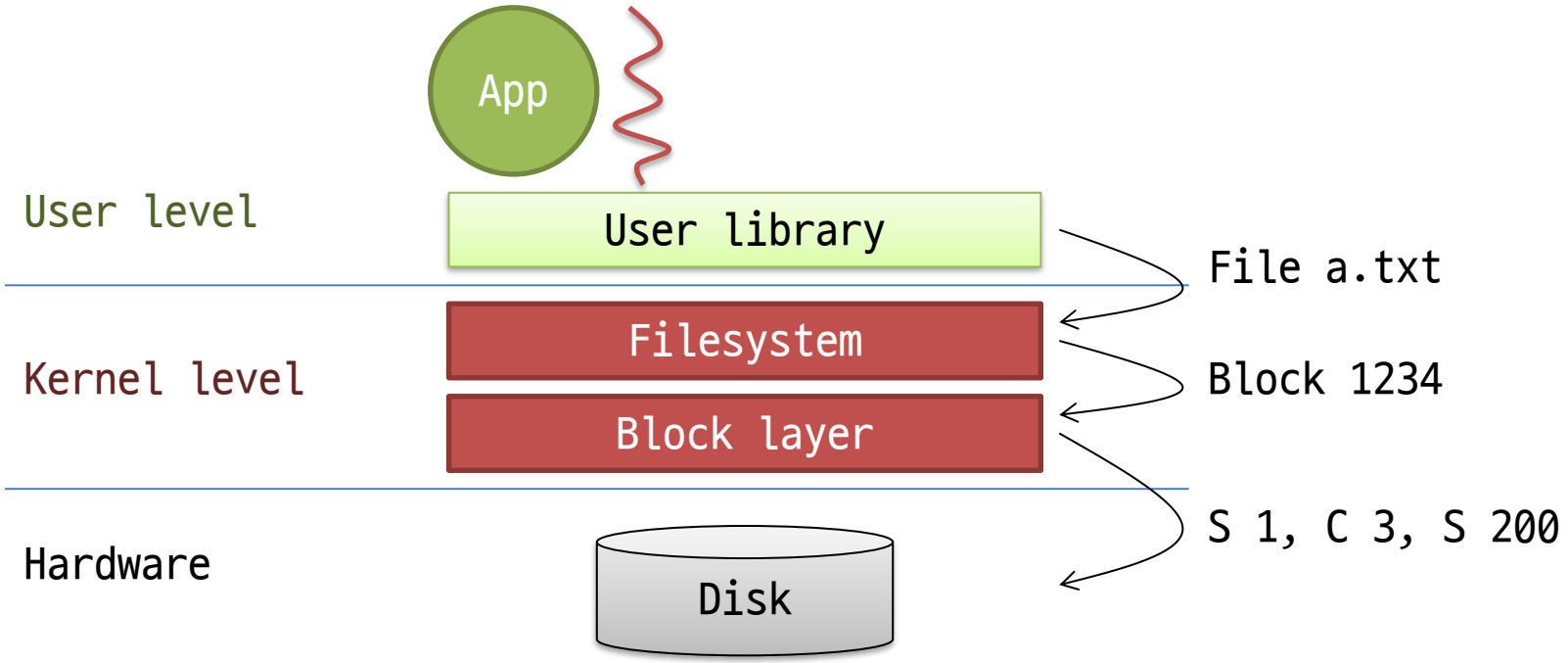
The job of the OS is to hide this mess from higher-level software

- Low-level device drivers (initiate a disk read, etc.)
- Higher-level abstractions (files, databases, etc.)

Managing Disks (2)

The OS may provide different levels of disk access

- Logical file (filename, block or record or byte #) by user library
- Disk logical block (disk block #) by filesystem
- Physical disk block (surface, cylinder, sector) by block layer



Managing Disks (3)

Specifying disk requests requires a lot of info:

- Cylinder #, surface #, track #, sector #, transfer size, etc.
- Older disks required the OS to specify all of this
- Therefore the OS needs to know all disk parameters

Modern disks are more complicated

- Not all sectors are the same size, sectors are remapped, etc.

Current disks provide a higher-level interface

- e.g., SCSI (Small Computer System Interface)
- The disks exports its data as a logical array of blocks [0..N-1]
- Disk maps logical blocks to cylinder/surface/track/sector
- Only need to specify the logical block # to read/write
- As a result, physical parameters are hidden from OS

Managing Disks (4)

Disk performance

- Seek: moving the disk arm to the correct cylinder
  - Depends on how fast disk arm can move (very slow)
- Rotation: waiting for the sector to rotate under head
  - Depends on rotation rate of disk (slow)
- Transfer: transferring data from surface into disk controller, sending it back to the host
  - Depends on density of bytes on disk (quick)

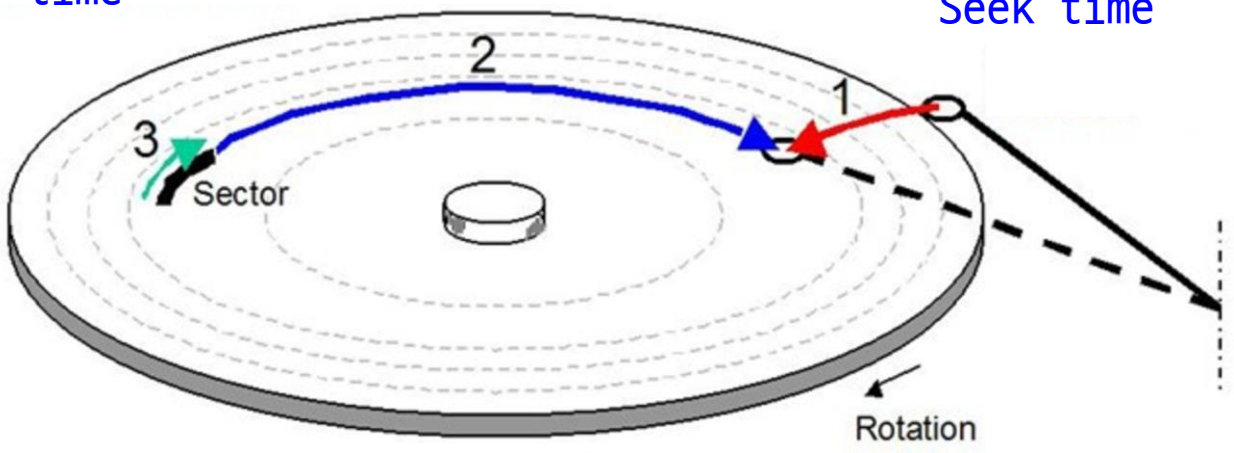
Managing Disks (5)

Disk performance

2. Disk rotation until the desired sector arrives under the head:
Rotational latency

3. Disk rotation until sector has passed under the head:
Data transfer time

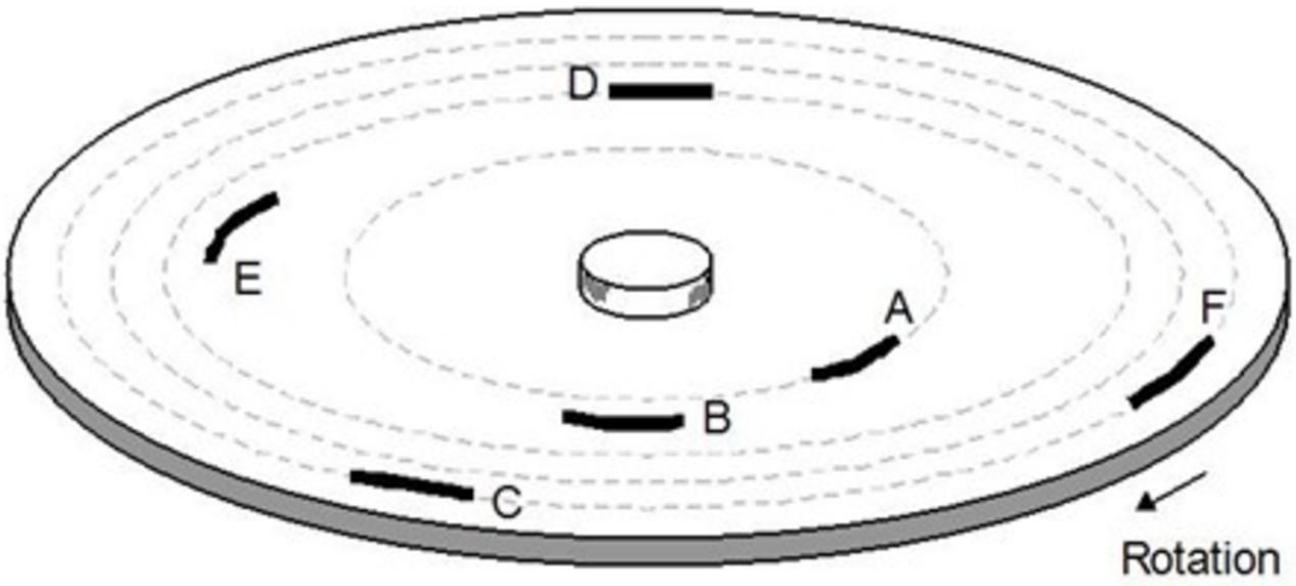
1. Head movement from current position to desired cylinder:
Seek time



Managing Disks (4)

Disk scheduling:

- Because seeks are so expensive
- When a read/write job is requested, the disk may currently be busy
- All pending jobs are placed in a disk queue
  - Could be scheduled to improve the utilization
- Disk scheduling increases the disk's bandwidth



Arrival order of access requests:

A, B, C, D, E, F

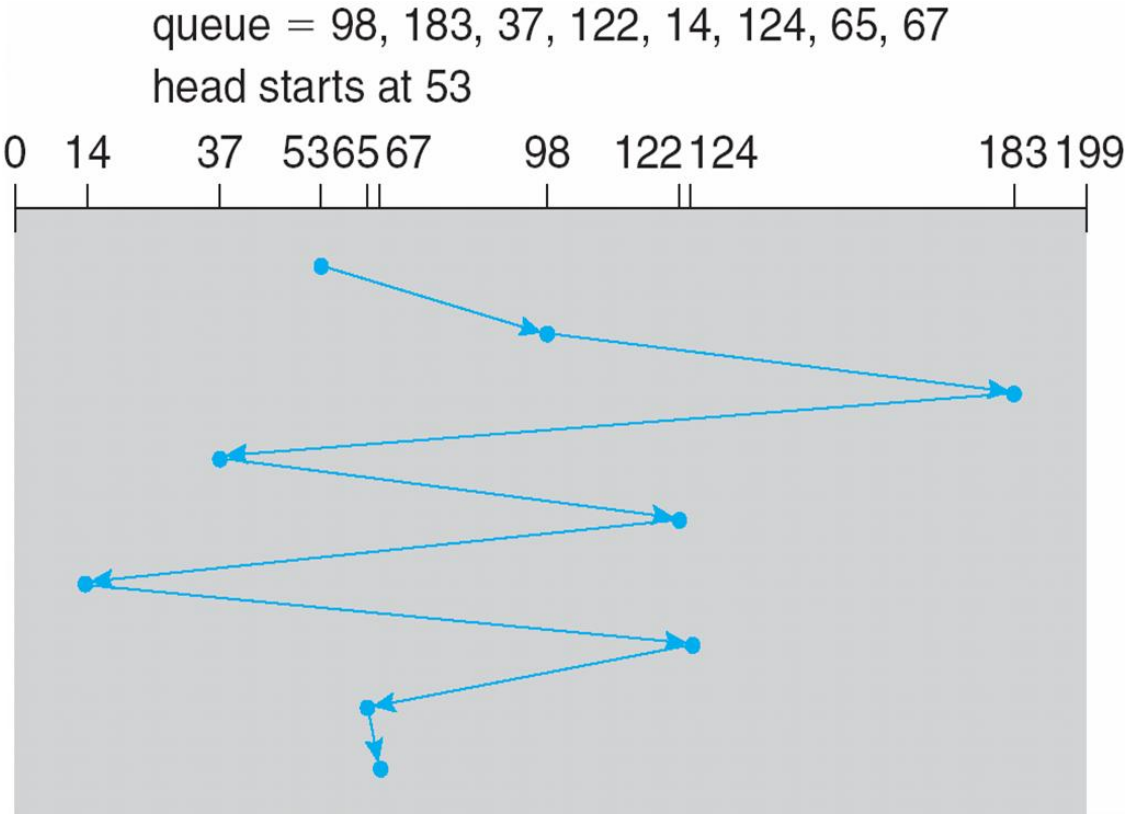
Possible out-of-order reading:

C, F, D, E, B, A

FCFS

FCFS (= do nothing)

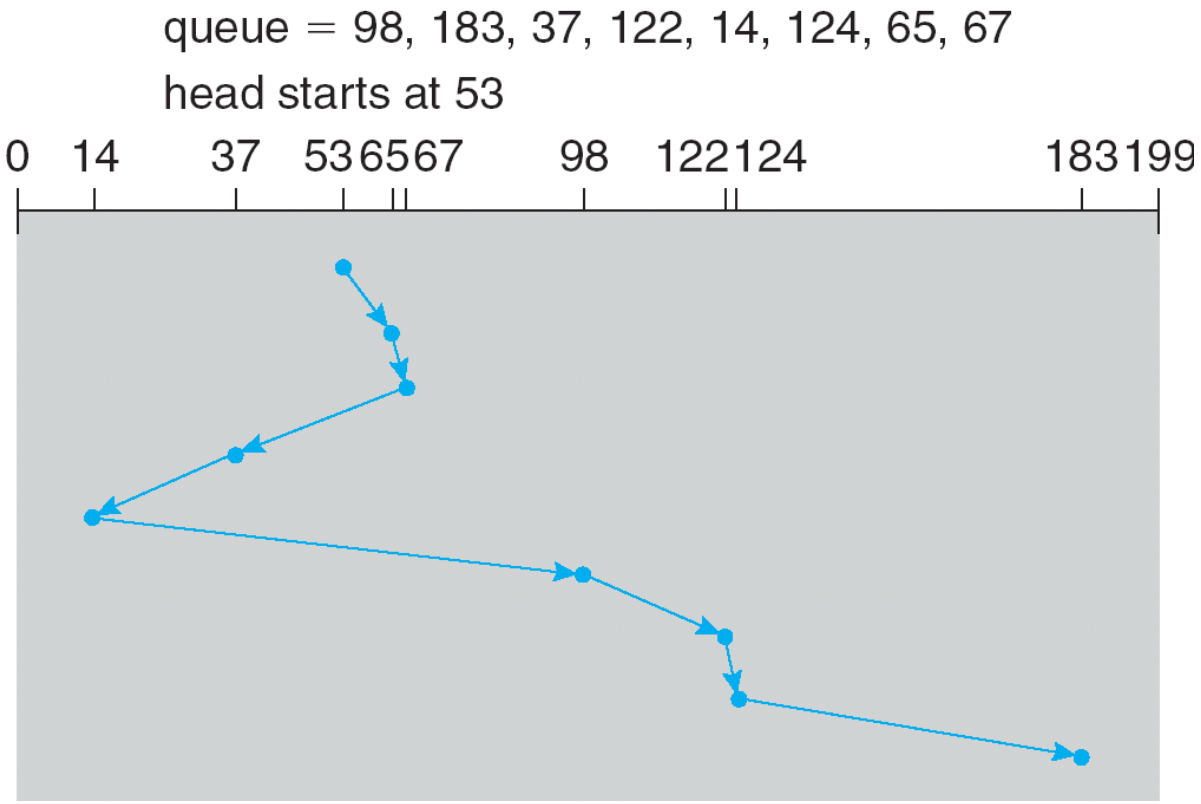
- Reasonable when load is low
- Long waiting times for long request queues



SSTF

Shortest seek time first

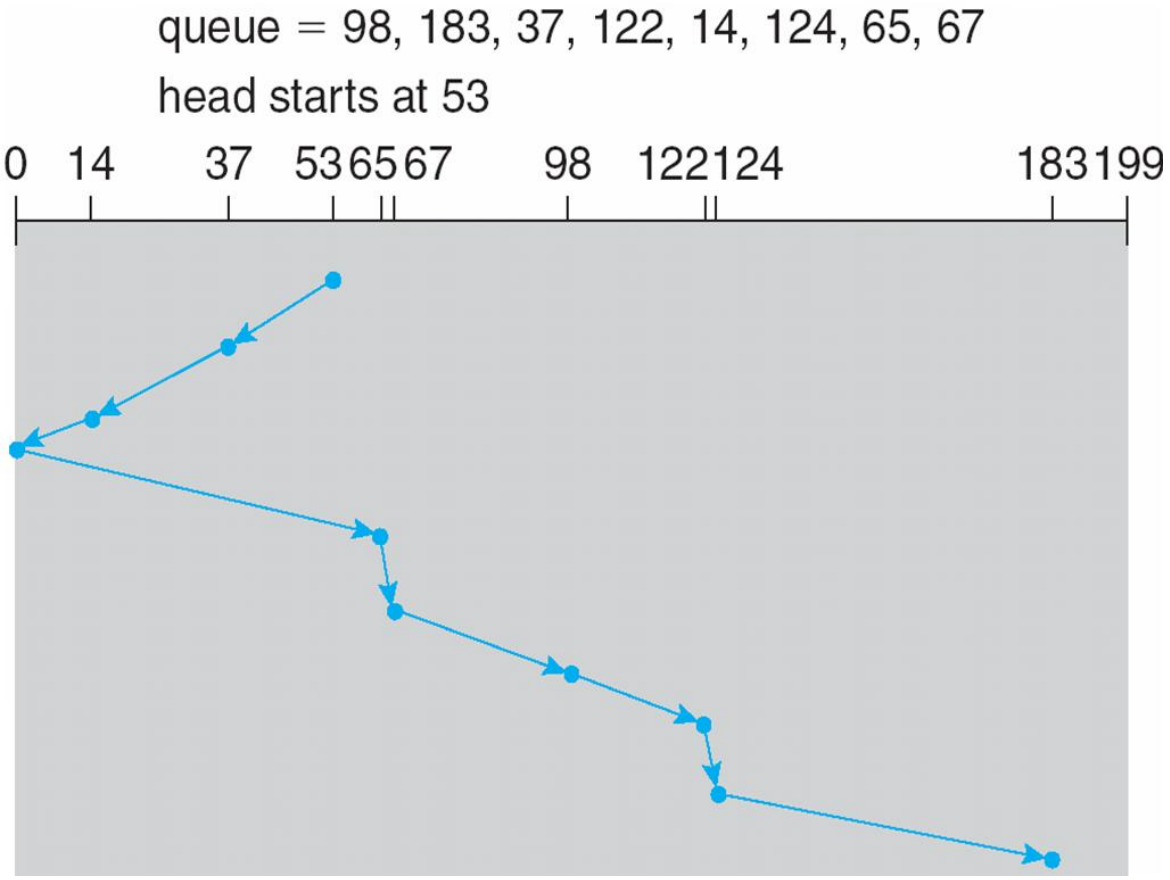
- Minimizes arm movement (seek time)
- Maximizes request rate
- Unfairly favors middle blocks
- May cause starvation of some requests



SCAN

Elevator algorithm

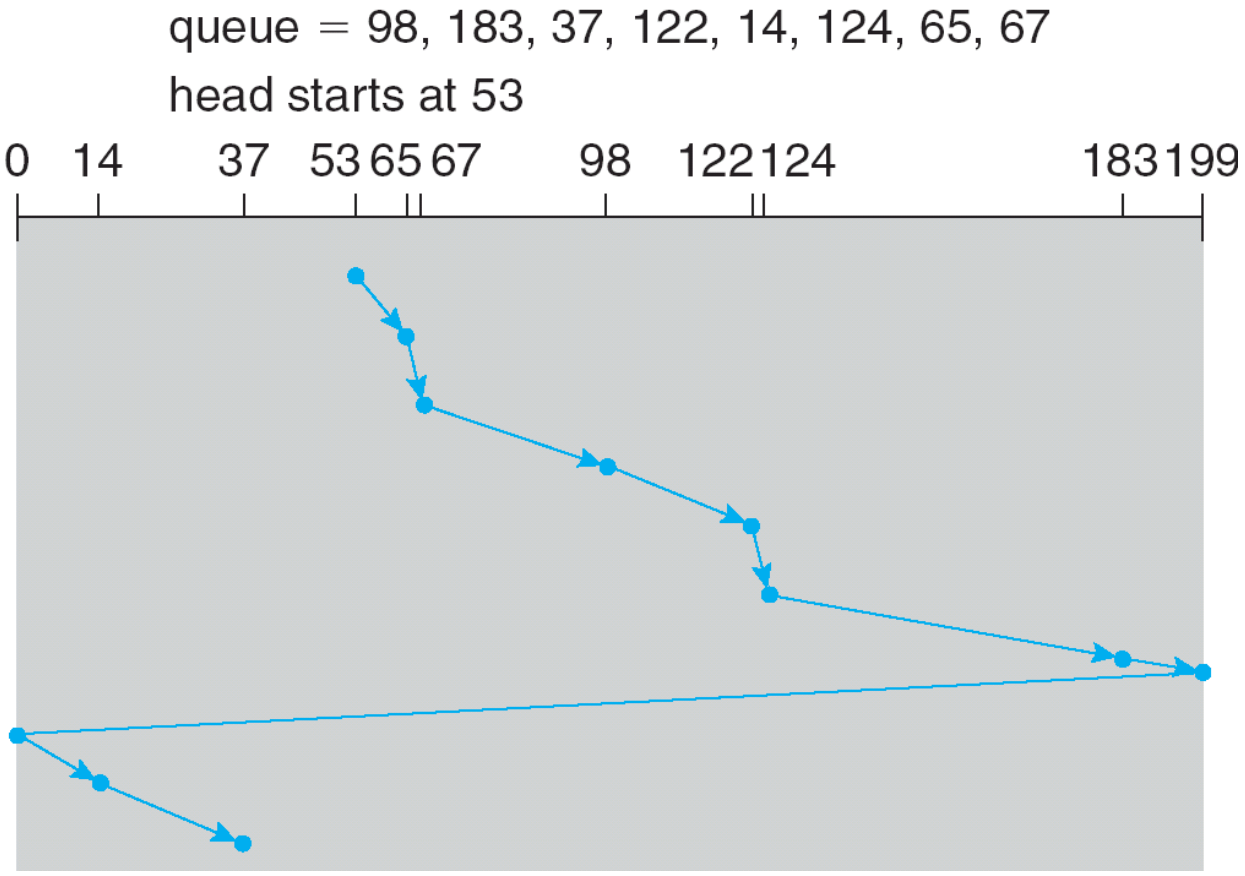
- Service requests in one direction until done, then reverse
- Skews wait times non-uniformly



C-SCAN

Circular SCAN

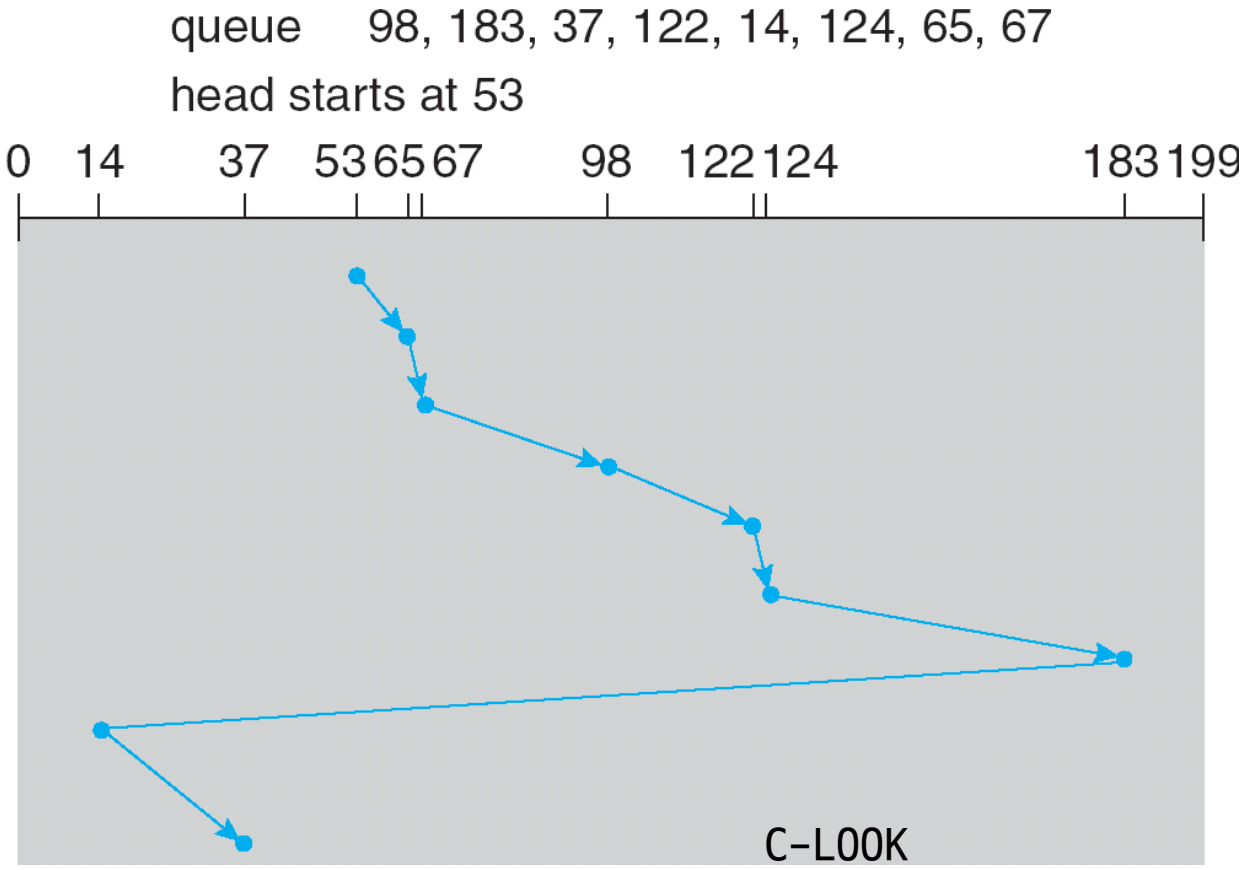
- Like SCAN, but only go in one direction (e.g. typewritters)
- Uniform wait times



LOOK / C-LOOK

LOOK / C-LOOK

- Similar to SCAN/C-SCAN
- But the arm goes only as far as the final request in each direction



Disk Scheduling

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
- Either SSTF or LOOK is a reasonable choice for the default algorithm
- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file allocation method

Modern disks often do the disk scheduling themselves

- Disks know their layout better than OS, can optimize better
- Ignores, undoes any scheduling done by OS

I/O Scheduler

I/O scheduler must

- Improve overall disk throughput
  - Merging requests to reduce the number of requests
  - Reordering and sorting requests to reduce disk seek time
- Prevent starvation
  - Submit requests before deadline
- Provide fairness among different processes
- Guarantee quality-of-service (QoS) requirement

Modern Disks

Intelligent controllers

- A small CPU + many kilobytes of memory
- They run a program written by the controller manufacturer

Intelligent features:

- Read-ahead: the current track
- Caching: frequently-used blocks
- Command queuing
- Request reordering: for seek and/or rotational optimality
- Request retry on hardware failure
- Bad block/track identification
- Bad block/track remapping: onto spare blocks and/or tracks