CSC139 Operating System Principles

Fall 2020, Part 4-1

Instructor: Dr. Yuan Cheng

Session Plan

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

Magnetic Disks

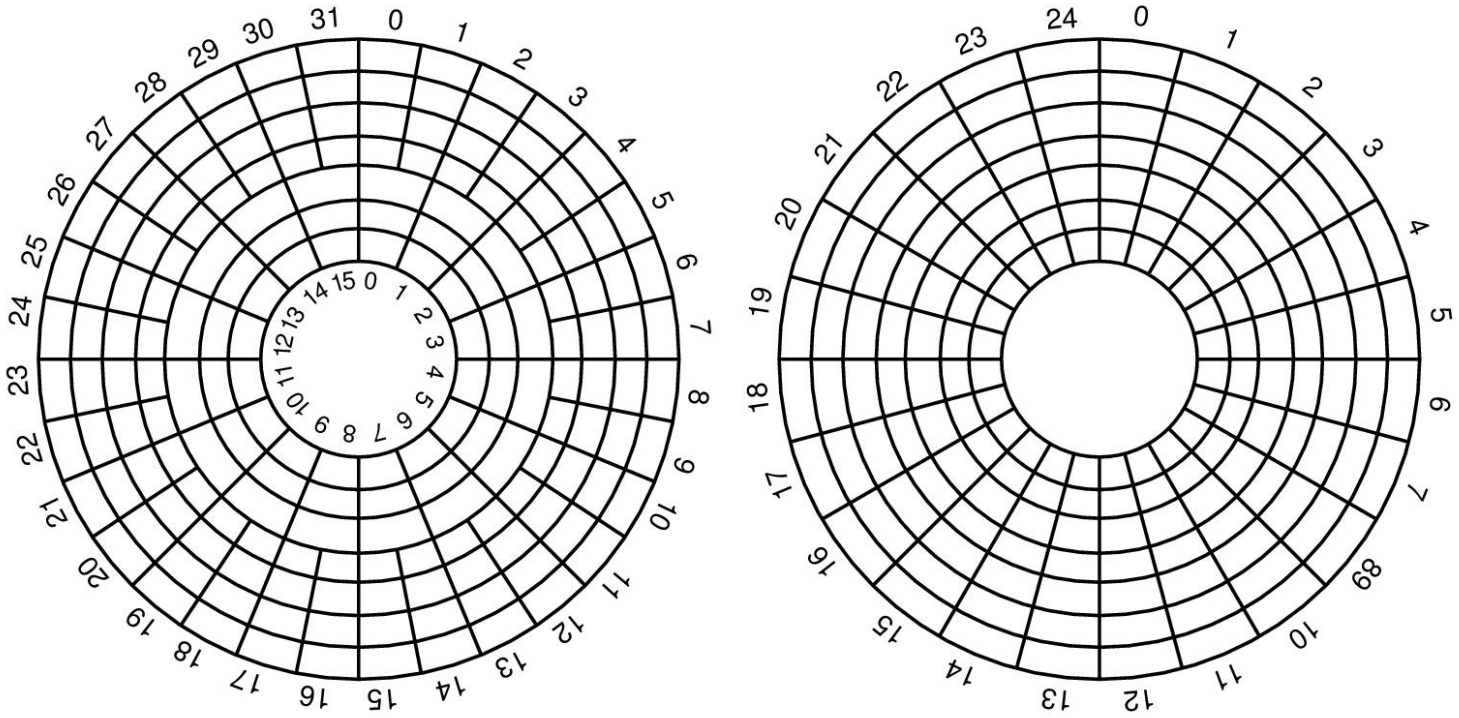
- Magnetic disks provide bulk of secondary storage of modern computers
- A magnetic disk has a sector-addressable address space
  - You can think of a disk as an array of sectors
  - Each sector (logical block) is the smallest unit of transfer
- Sectors are typically 512 or 4096 bytes
- Main operations
  - Read from sectors (blocks)
  - Write to sectors (blocks)

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 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 should be easy
    - Except for bad sectors
    - Non-constant # of sectors per track via constant angular velocity

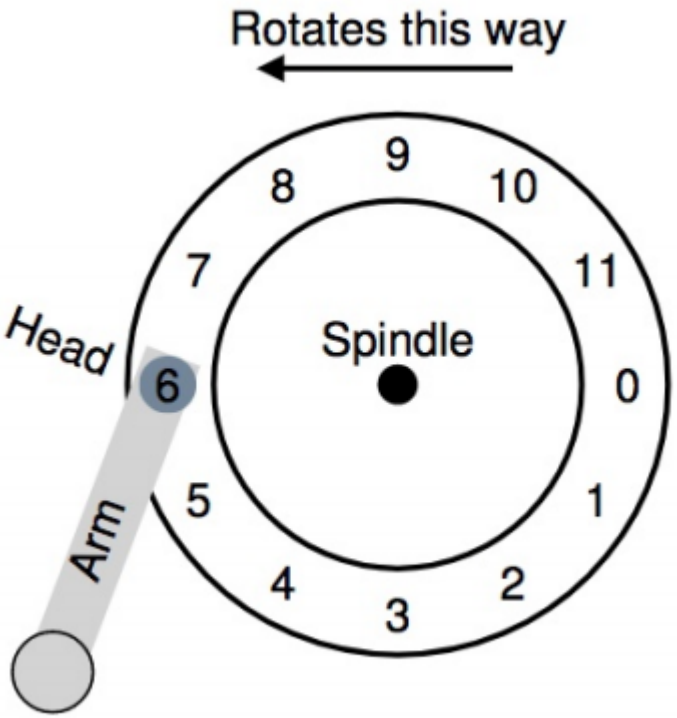
Non-uniform #sectors / track

- Reduce bit density per track for outer layers (Constant Linear Velocity, typically HDDs)
- Have more sectors per track on the outer layers, and increase rotational speed when reading from outer tracks (Constant Angular Velocity, typically CDs, DVDs)

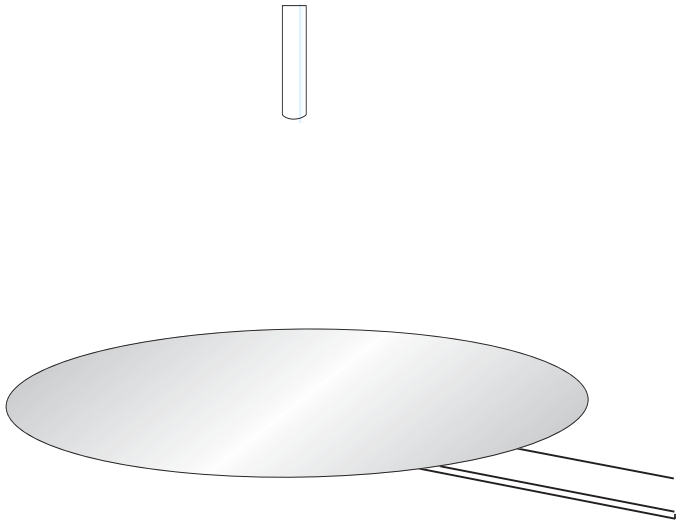


Internals of Hard Disk Drive (HDD)

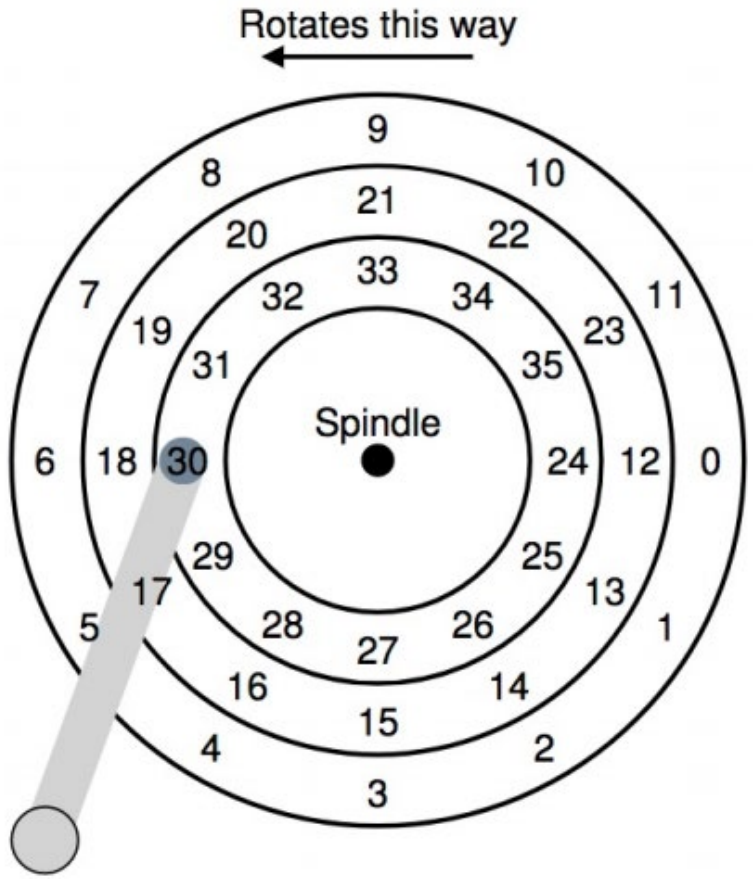
A single track + an arm + a head



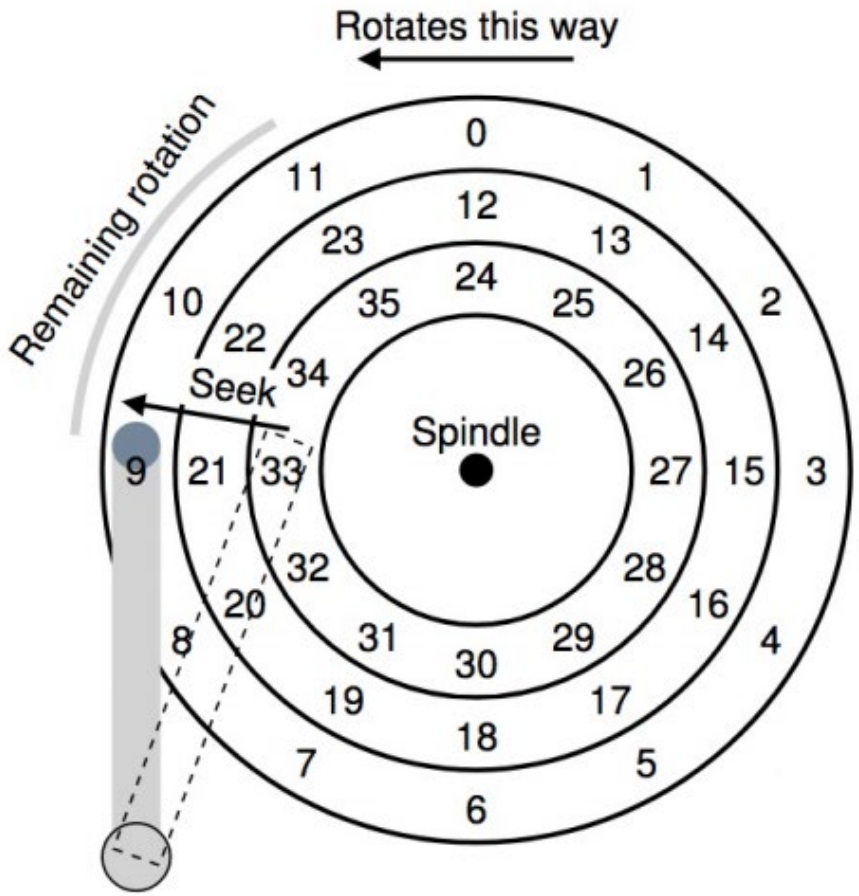
Moving-head Disk Mechanism



Read Sector 0



Read Sector 0



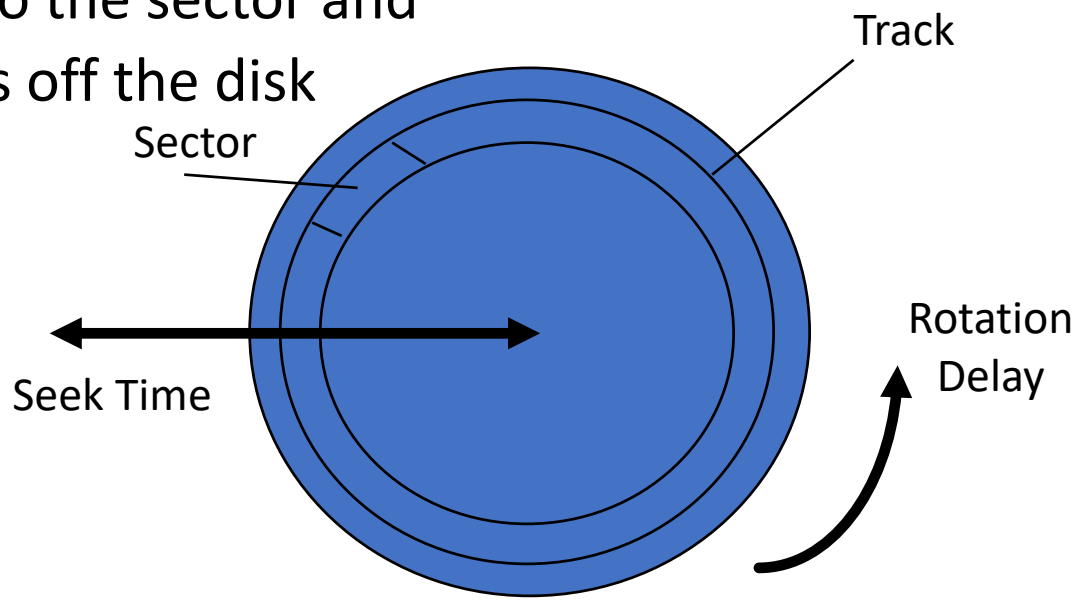
1. Seek for right track
2. Rotate (sector 9 → 0)
3. Transfer data (sector 0)

Inside of a Hard Disk Drive

- https://www.youtube.com/watch?v=9eMWG3fwiEU

Disk Overhead

- To read from disk, we must specify:
  - cylinder #, surface #, sector #, transfer size, memory address
- Transfer time includes:
  - Seek time: to get to the track
  - Latency time: to get to the sector and
  - Transfer time: get bits off the disk



Disk Performance

- I/O latency of disks
  - $L_{I/O} = L_{seek} + L_{rotate} + L_{transfer}$
- Disk access latency at millisecond level

Seek, Rotate, Transfer

- Seek may take milliseconds (ms)
  - Entire seek often takes 3 – 9 ms
- Rotation per second (RPM)
  - 7200 RPM is common nowadays
  - 15000 RPM is high end
  - Old computers may have 5400 RPM disks
  - 1 / 7200 RPM = 1 min / 7200 rotations = 1 sec / 120 rotations = 8.3 ms / rotation => it may take 4.2 ms on average to rotate to target (0.5 * 8.3 ms)
- Transfer is relatively fast
  - 100+MB/s for SATA I, up to 600MB/s for SATA III
  - 1s / 100 MB = 10 ms / MB = 4.9 us/sector (assuming 512-byte sector)

Disk Performance Calculation

- RPM 7200
- Avg seek 4.16ms
- Max transfer 500MB/s
- How long does an average 4KB read take?
  - Transfer = 1sec/500MB * 4KB * 1,000,000us/1sec = 8 us
  - Latency = 4.16 ms (avg seek) + 4.2 ms (avg rotate) + 8 us = 8.368 ms

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

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

The First Commercial Disk Drive

1956
IBM RAMDAC computer included the IBM Model 350 disk storage system

5M (7 bit) characters
50 x 24” platters
Access time = < 1 second

Modern Disks

	Barracuda 180	**Cheetah X15 36LP**
Capacity	181GB	36.7GB
Disk/Heads	12/24	4/8
Cylinders	24,247	18,479
Sectors/track	~609	~485
Speed	7200RPM	15000RPM
Latency (ms)	4.17	2.0
Avg seek (ms)	7.4/8.2	3.6/4.2
Track-2-track(ms)	0.8/1.1	0.3/0.4

Solid-State Disks

- Nonvolatile memory used like a hard drive
  - Many technology variations
- Can be more reliable than HDDs
- More expensive per MB
- Maybe have shorter life span
- Less capacity
- But much faster
  - Very fast reads
  - Writes are slower – need a slow erase cycle (cannot overwrite directly)
- 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
- Common technologies are LTO-{3,4,5} and T10000

Disks vs. Memory

- Smallest write: sector
- Atomic write = sector
- Random access: 5ms
  - not on a good curve
- Sequential access: 200MB/s
- Cost $.002MB
- Crash: doesn't matter ("non-volatile")

- (usually) bytes
- byte, word
- 50 ns
  - faster all the time
- 200-1000MB/s
- $.10MB
- contents gone ("volatile")

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
- Access time has two major components
  - Seek time is time to move the heads to the cylinder containing the desired sector
  - Rotational latency is additional time waiting to rotate the desired sector to the disk head
- 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, memory 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 means work must queue
  - Optimization algorithms only make sense when a queue exists

Disk Scheduling (cont.)

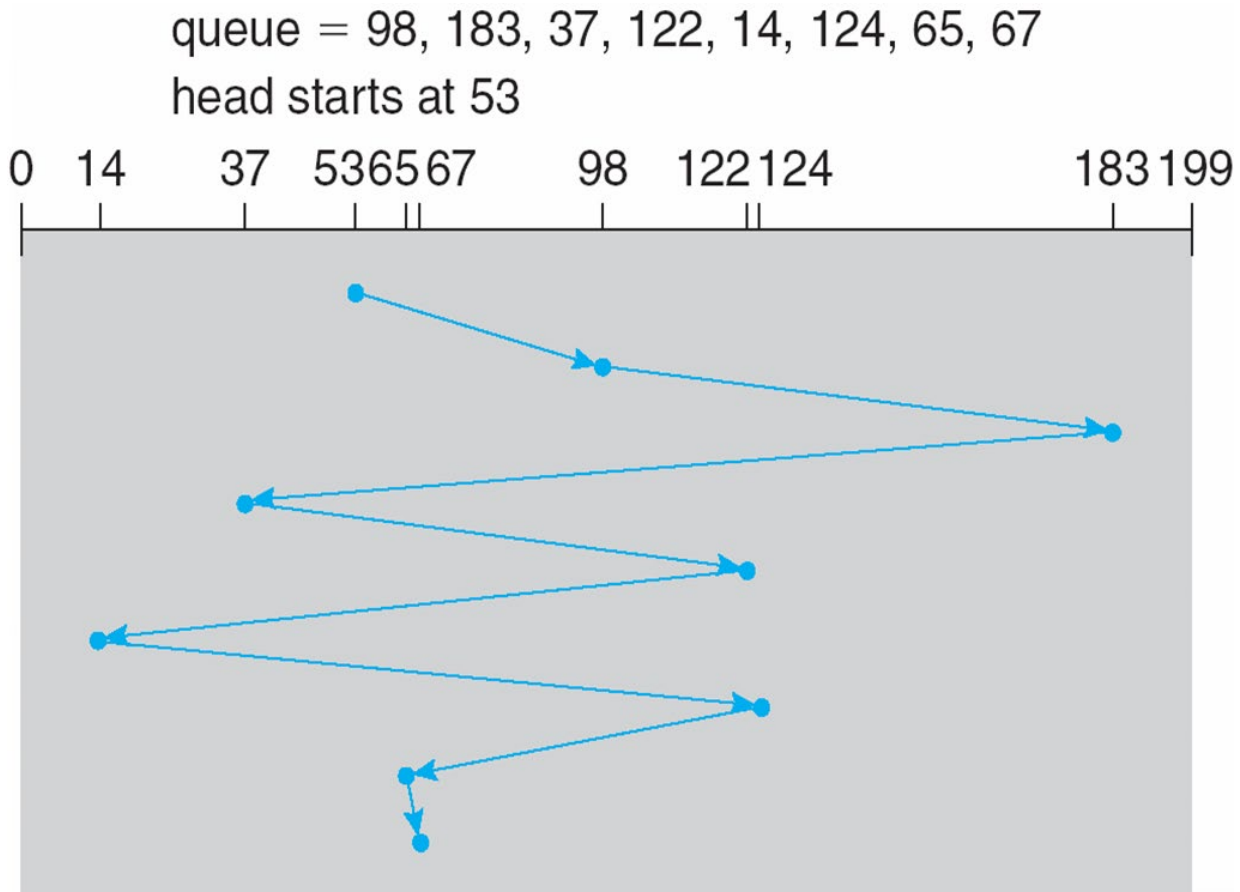
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying “depth”)
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199)

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

Head pointer 53

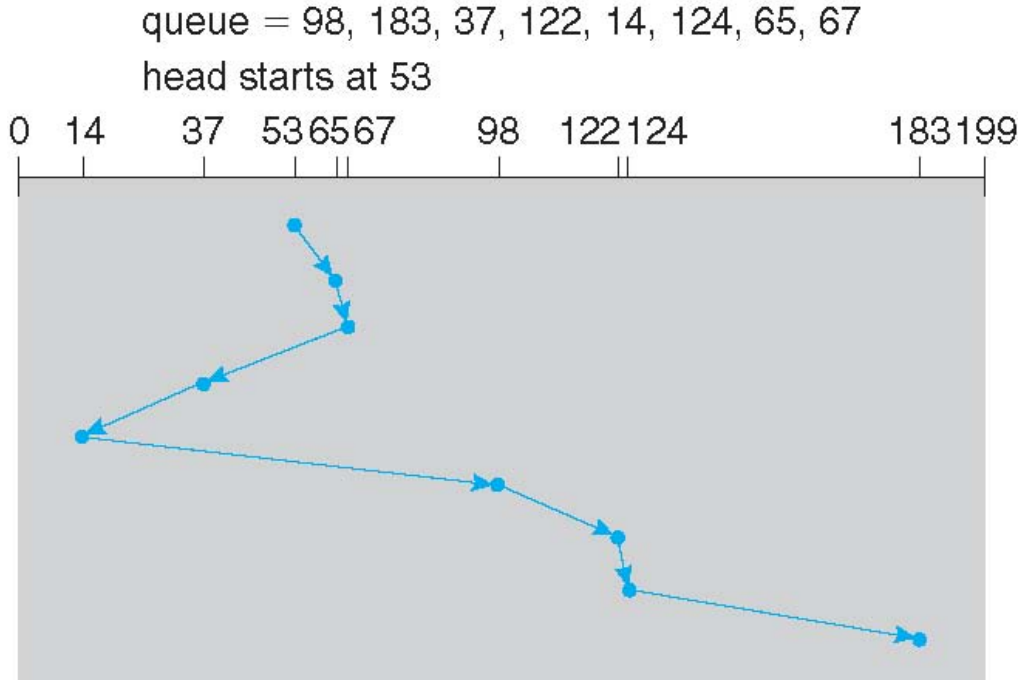
FCFS

- Illustration shows total head movement of 640 cylinders



SSTF

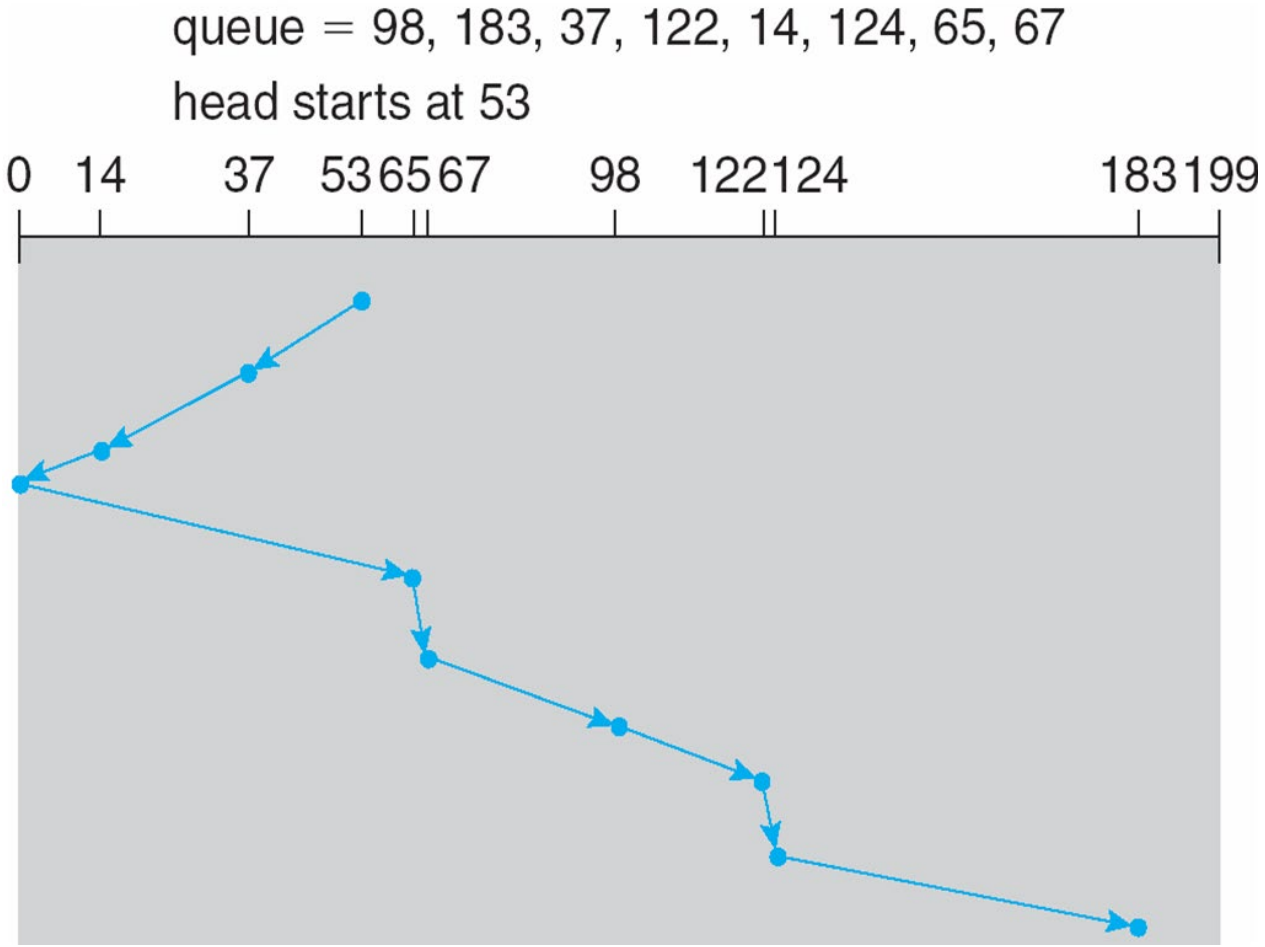
- Shortest Seek Time First selects the request with the minimum seek time from the current head position
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests
- Illustration shows total head movement of 236 cylinders



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.
- SCAN algorithm sometimes called the elevator algorithm
- Illustration shows total head movement of 208 cylinders
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

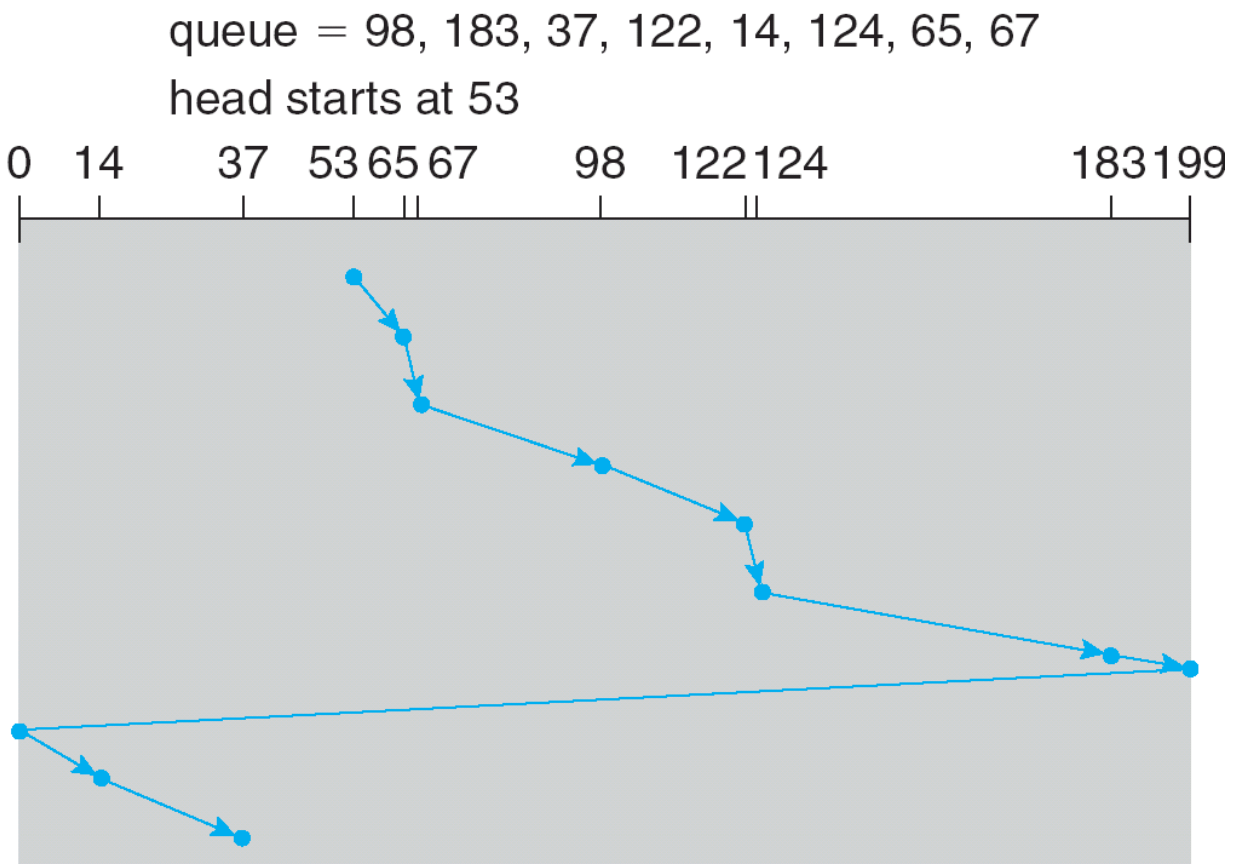
SCAN (cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

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
- Total number of cylinders?

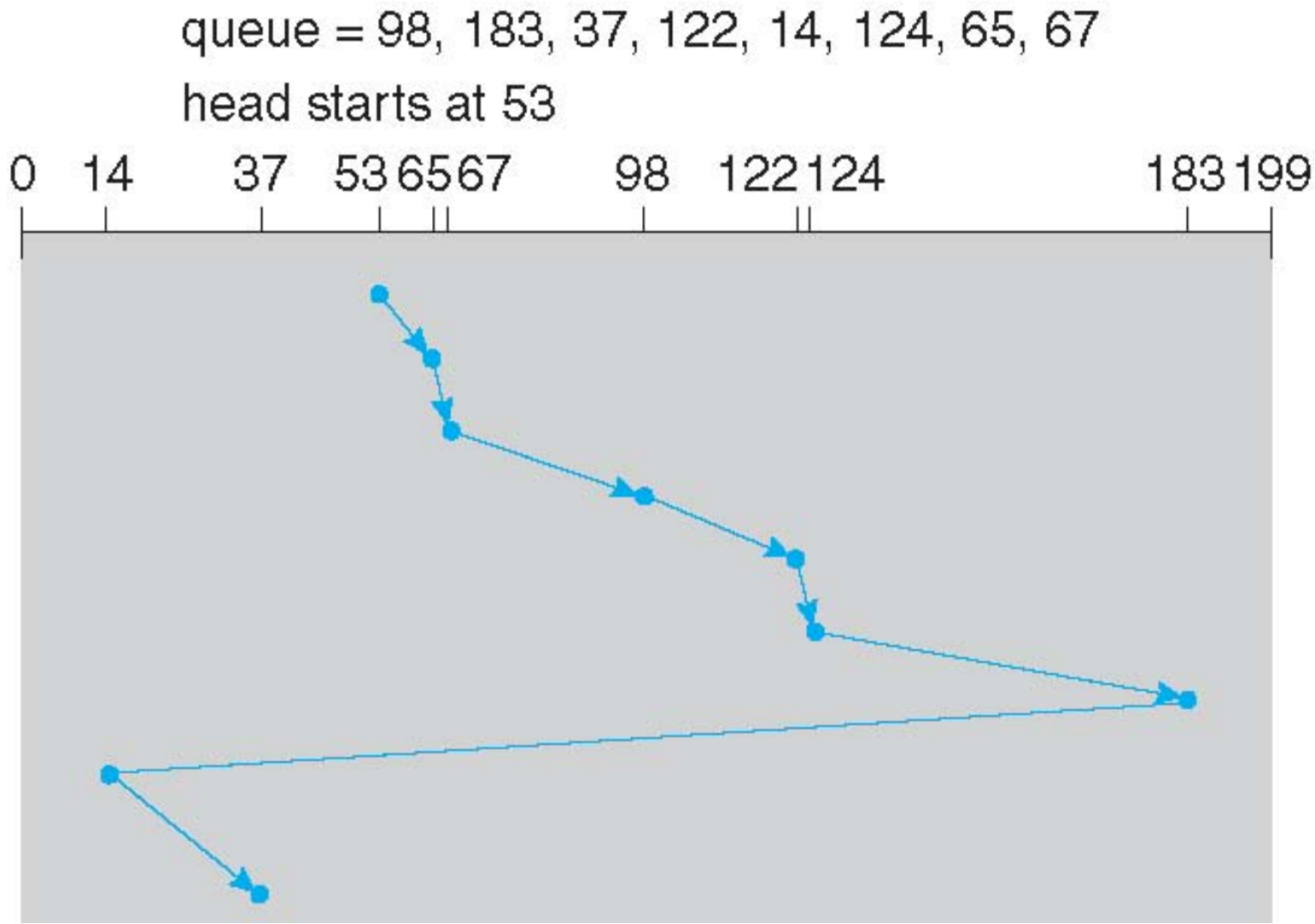
C-SCAN (cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

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?

C-LOOK (cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Disk Scheduling Algorithms – an Interactive Example

- https://www.cs.usask.ca/faculty/makaroff/cgi-bin/disk_sched.pl

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
- 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
- What about rotational latency?
  - Difficult for OS to calculate

RAID Structure

- RAID – redundant array of independent disks
  - multiple disk drives provides reliability via redundancy
- Increases the mean time to failure
- Mean time to repair – exposure time when another failure could cause data loss
- Mean time to data loss based on above factors
- If mirrored disks fail independently, consider disk with 1300,000 mean time to failure and 10 hour mean time to repair
  - Mean time to data loss is $100,000^2 / (2 * 10) = 500 * 10^6$ hours, or 57,000 years!
- Frequently combined with NVRAM to improve write performance
- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively

RAID (cont.)

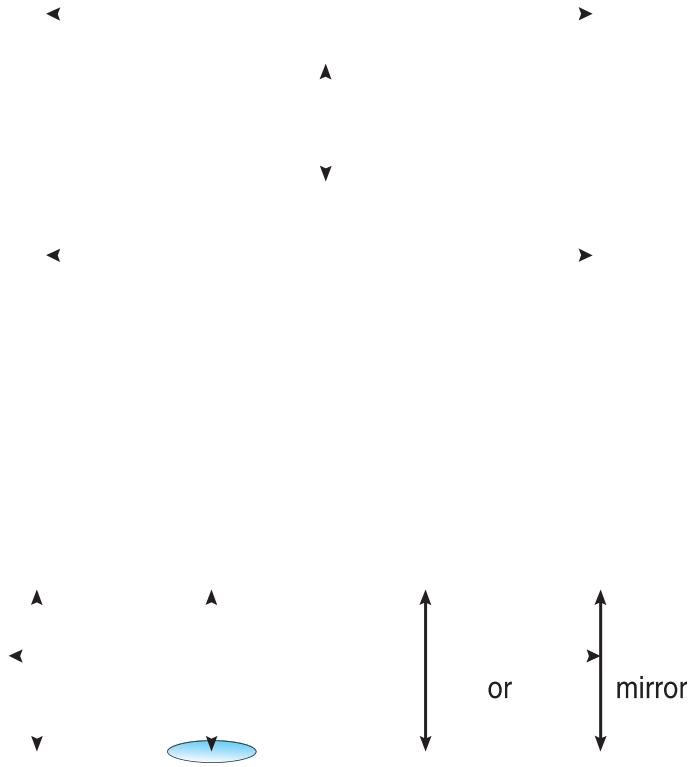
- Disk striping uses a group of disks as one storage unit
- RAID is arranged into six different levels
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
  - Mirroring or shadowing (RAID 1) keeps duplicate of each disk
  - Striped mirrors (RAID 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
- RAID within a storage array can still fail if the array fails, so automatic replication of the data between 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

RAID Structure

- RAID – multiple disk drives provide reliability via redundancy
- Disk striping uses a group of disks as one storage unit
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
  - Mirroring keeps duplicate of each disk
  - Block interleaved parity uses much less redundancy
- RAID is arranged into six different levels

RAID Levels

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



Other Features

- Regardless of where RAID implemented, other useful features can be added
- Snapshot is a view of file system before a set of changes take place (i.e. at a point in time)
  - More in Ch 12
- 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 if a disk fails to replace the failed disk and rebuild the RAID set if possible
  - Decreases mean time to repair

Summary

- Disks are slow devices relative to CPUs.
- For most OS features, we are very concerned about efficiency.
- For I/O systems, and disk, in particular, it is worthwhile to complicate and slow down the OS if we can gain improvement in I/O times.

Exit Slips

- Take 1-2 minutes to reflect on this lecture
- On a sheet of paper write:
  - One thing you learned in this lecture
  - One thing you didn’t understand

Next class

- We will discuss:
- Reading assignment:

Acknowledgment

- The slides are partially based on the ones from
  - The book site of *Operating System Concepts (Tenth Edition)*: http://os-book.com/