CS370 Operating Systems

Colorado State University
Yashwant K Malaiya

Fall 2016 Lecture 36

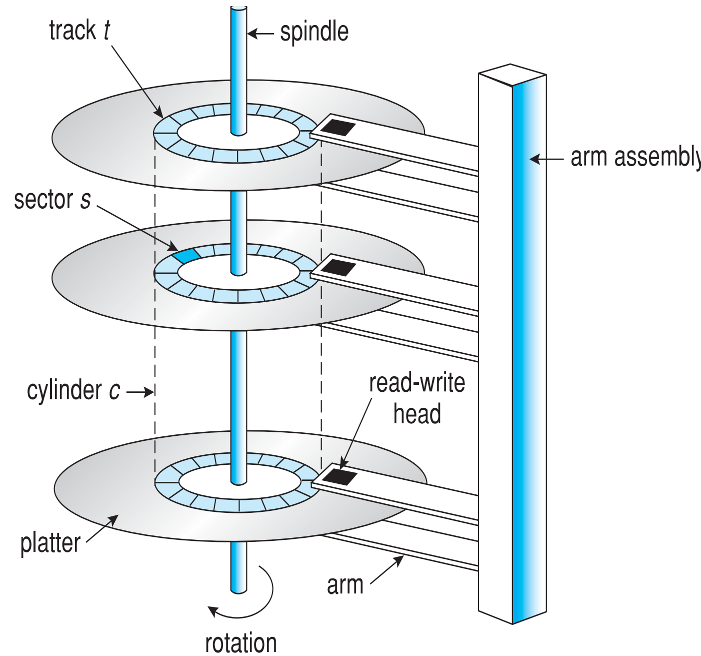
Mass Storage

Slides based on

- **Text by Silberschatz, Galvin, Gagne**
- **Various sources**

FAQ

- Why are tapes still used?
- Is it common to have multiple platters in HDs? How many?
- Cylinders, tracks, sectors
- Does average seek time increase when there is fragmentation on the HD?
- Rotational latency? Formulas? again
- How is data accessed in SSD if there are no moving parts?
- Why SSD is not universal?
- SSD: wear leveling: is it done now?
- SSD have sectors?
- Various buses/protocols: SCSI etc
- NAS vs SAN: shared vs dedicated (faster) network for servers
- RAID? soon

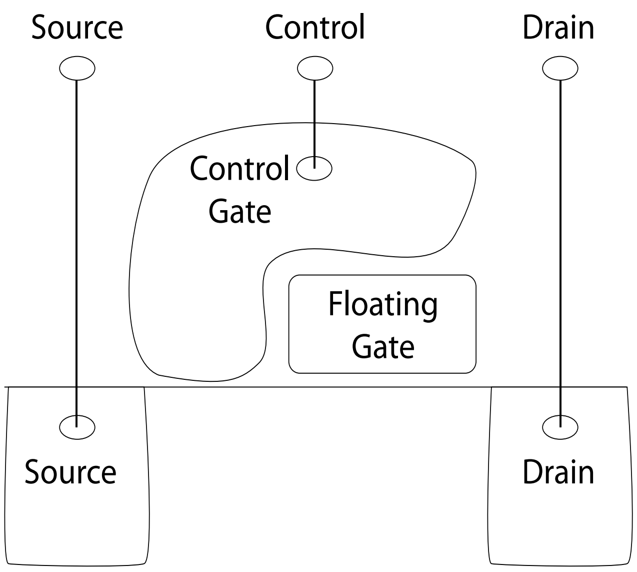


Access time= seek time + rotational latency

Colorado State University

Flash Memory

- Writes must be to “clean” cells; no update in place
  - Large block erasure required before write
  - Erasure block: 128 – 512 KB
  - Erasure time: Several milliseconds
- Write/read page (2-4KB)
  - 50-100 usec
- Why are random writes so slow?
  - Random write: 2000/sec
  - Random read: 38500/sec



Colorado State University

Hard Disk Performance

- **Average access time** = average seek time + average latency
- Average I/O time = average access time + transfer time + 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) + transfer time+ 0.1ms
  - Transfer time = amount to transfer / transfer rate 4KB / 1Gb/s = 4x8K/G = 0.031 ms
  - Average I/O time for 4KB block = 9.27ms + .031ms = 9.301ms

Disk Formatting

- Low-level formatting marks the surfaces of the disks with markers indicating the start of a recording block (sector markers) and other information by the disk controller to read or write data.
- Partitioning divides a disk into one or more regions, writing data structures to the disk to indicate the beginning and end of the regions. Often includes checking for defective tracks/sectors.
- High-level formatting creates the file system format within a disk partition or a logical volume. This formatting includes the data structures used by the OS to identify the logical drive or partition's contents.

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 (between cylinders)
- 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.)

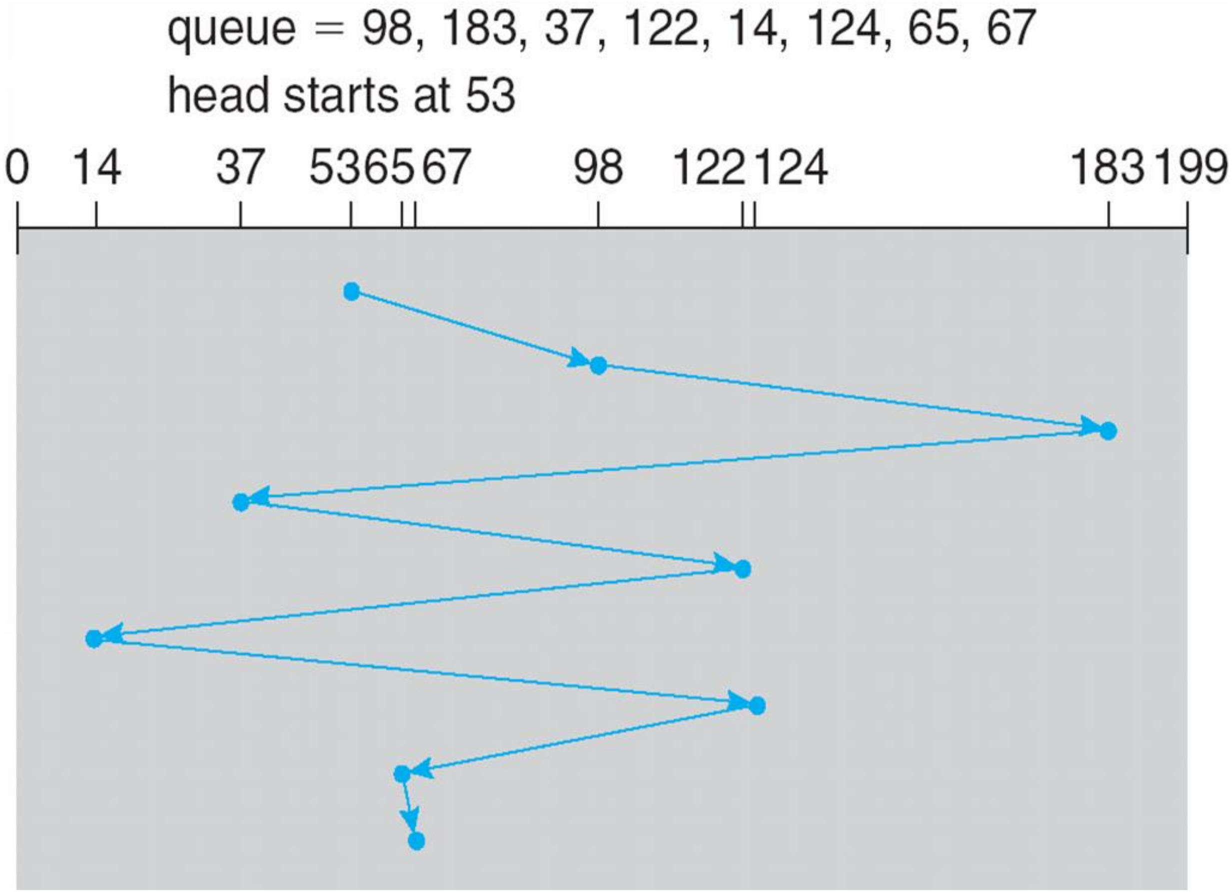
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying "depth")
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (cylinders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

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

  Head pointer 53 (head is at cylinder 53)

FCFS (First come first served)

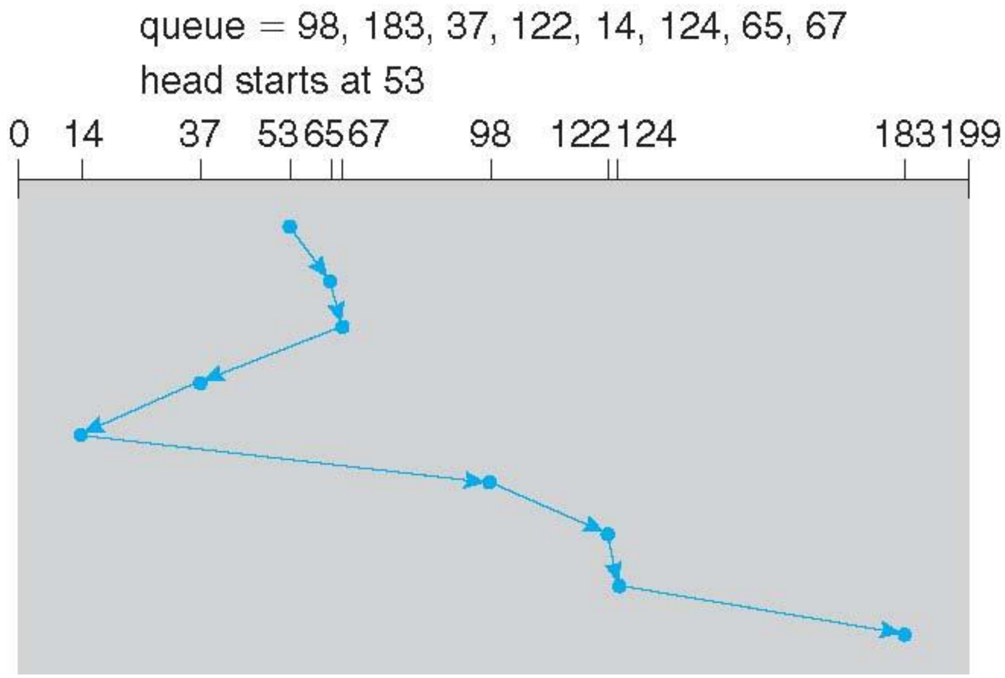
Illustration shows total head movement



Total seek time **640** cylinders

SSTF Shortest Seek Time First

- **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
- total head movement of **236** cylinders

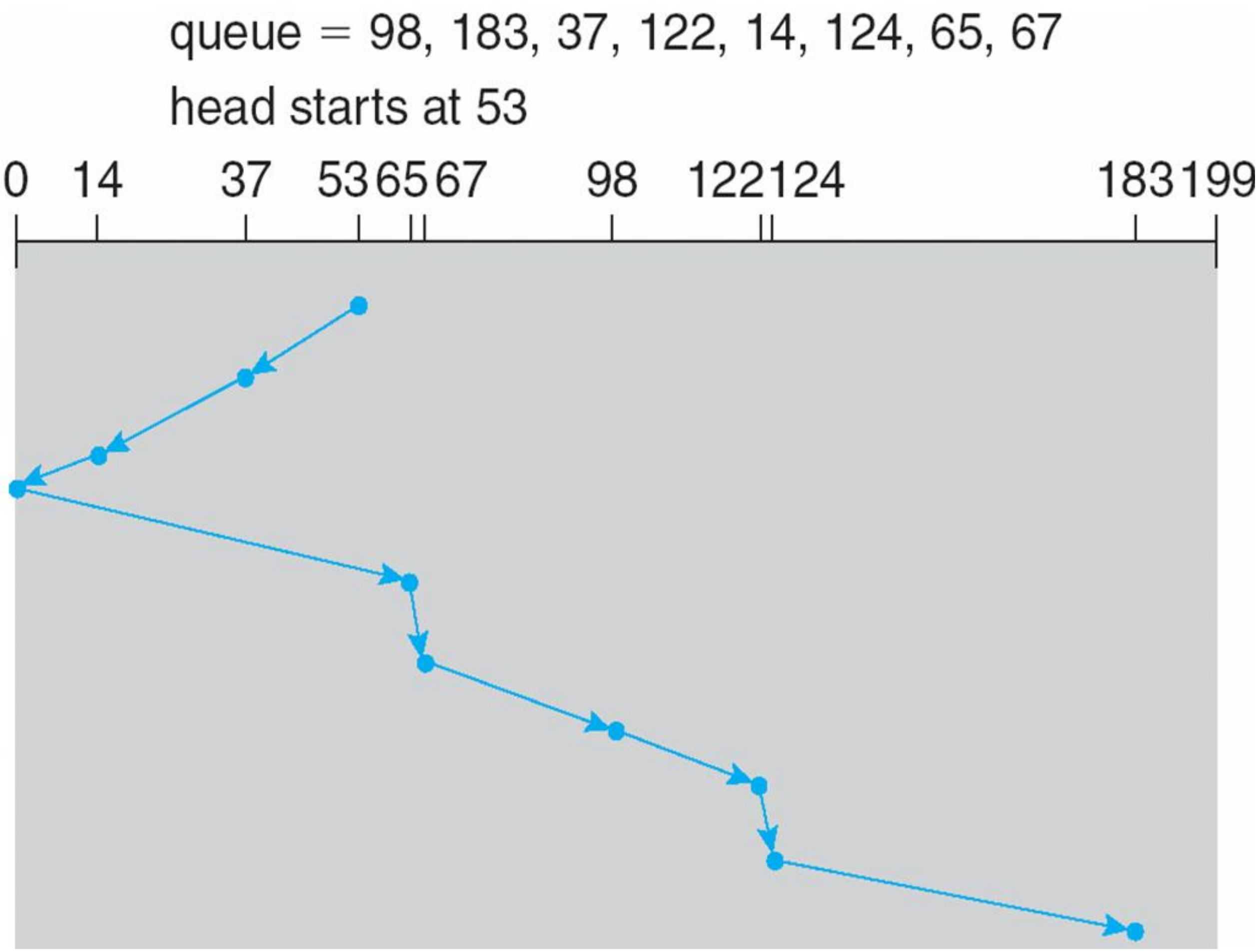


Colorado State University

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**
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

SCAN (Cont.)

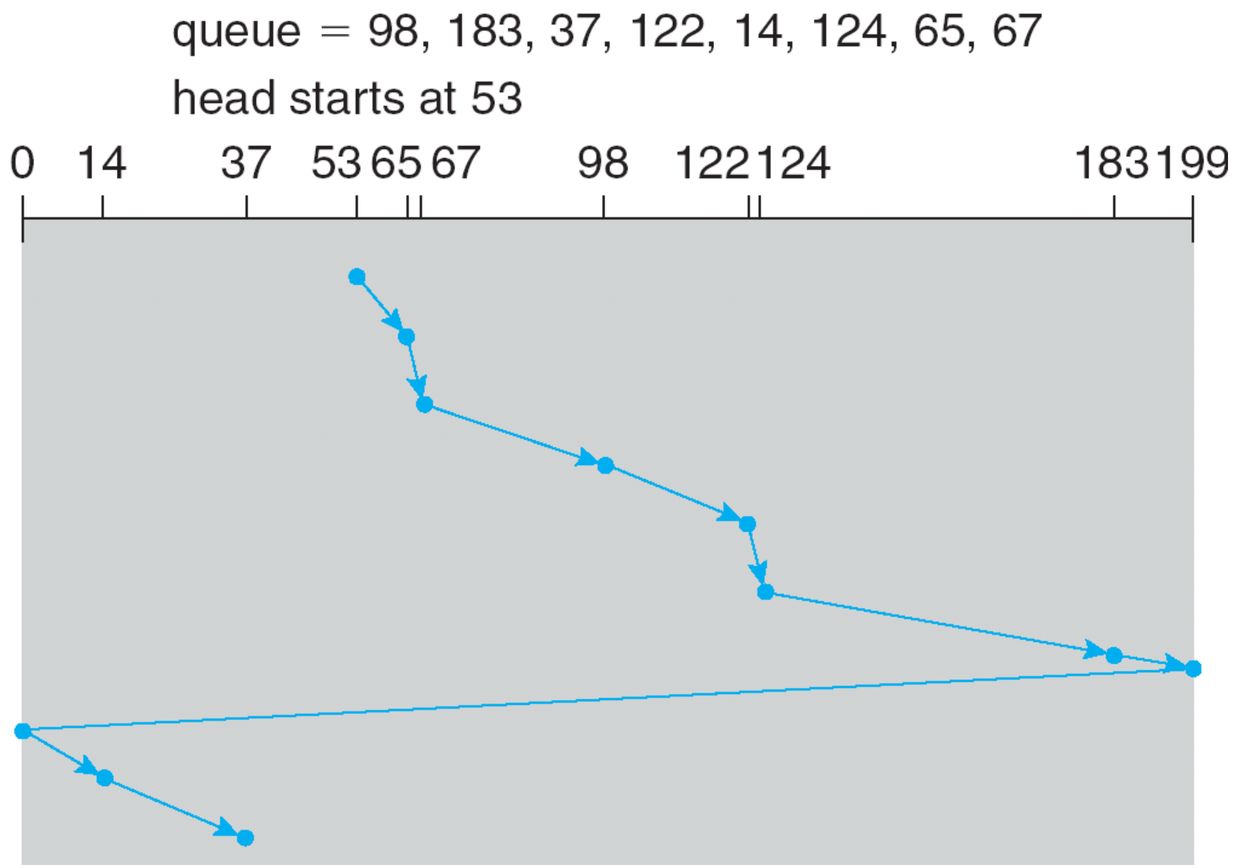


Total 53+ 183= **236** cylinders

C-SCAN (circular 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? 183

C-SCAN (Cont.)



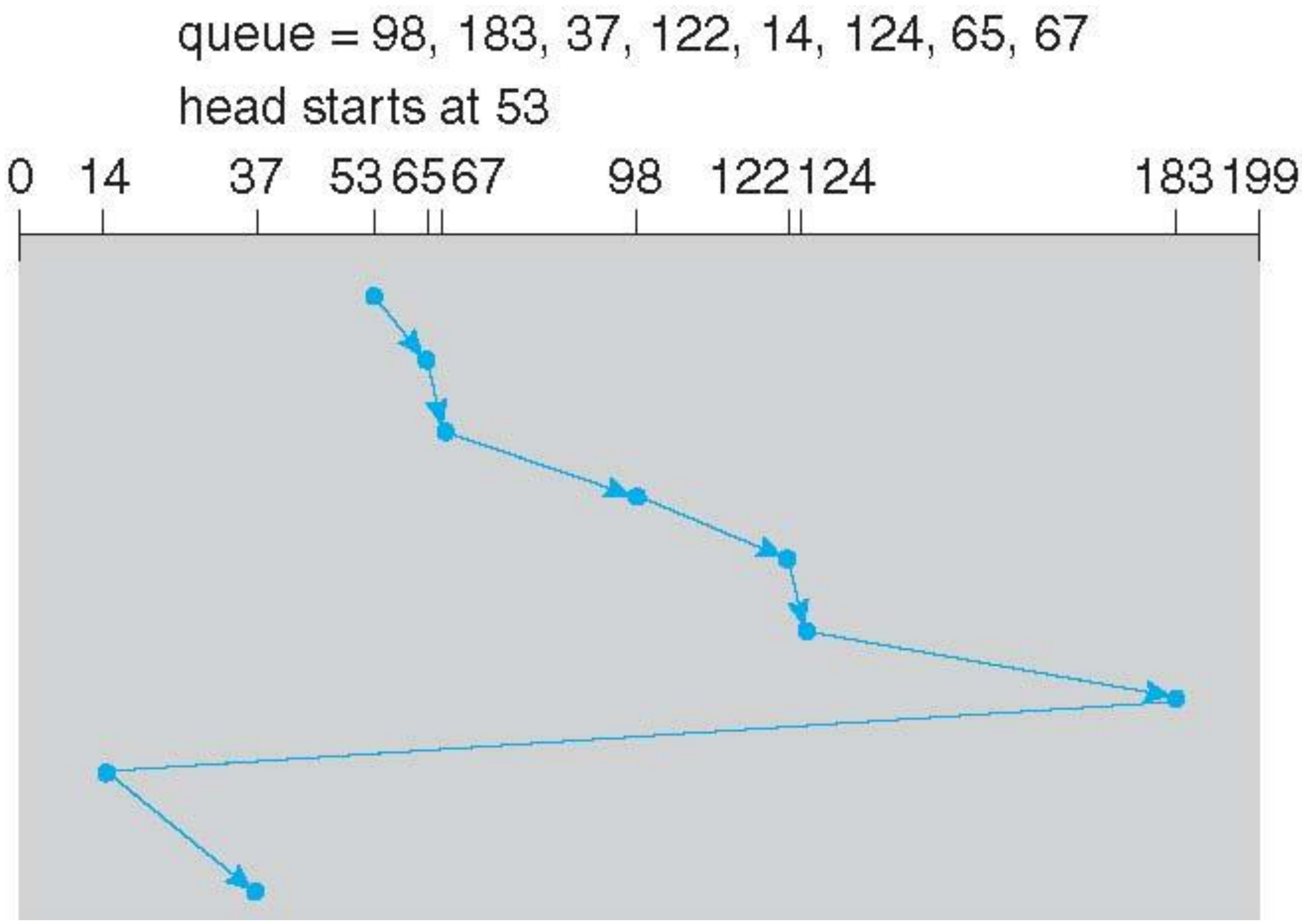
Total (199-53)+ 199+37= **382** cylinders

Colorado State University

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.)



Colorado State University

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
- Either SSTF or LOOK is a reasonable choice for the default algorithm
- What about rotational latency?
  - Difficult for OS to calculate

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 (sector number), plus data, plus error correction code (**ECC**)
  - 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

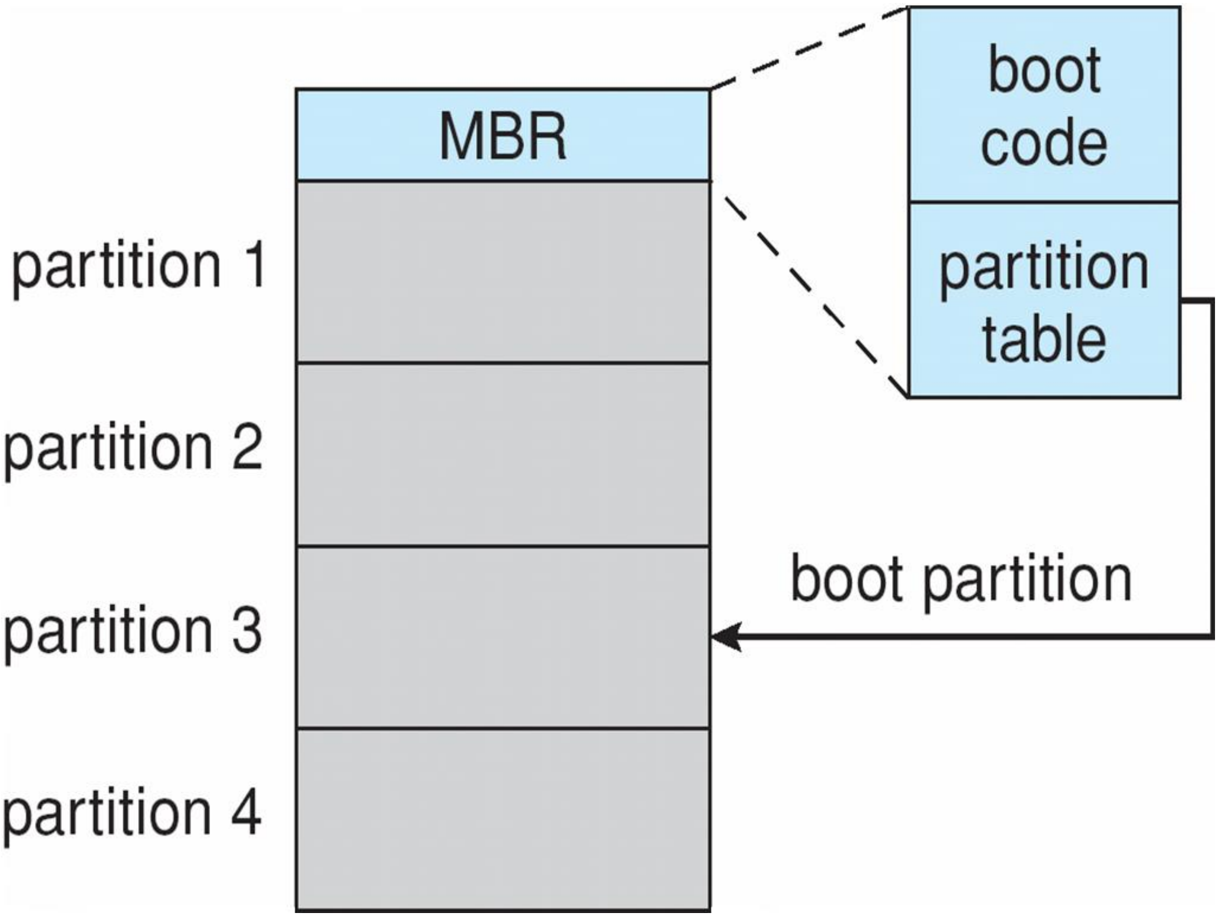
Block: a logical unit addressable by OS.
One or more sectors

Disk Management (Cont.)

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

Boot disk: has a boot partition

Booting from a Disk in Windows



MBR: Master boot record
Kernel loaded from boot partition

RAID Structure

- RAID – redundant array of inexpensive 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 100,000 hour 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!

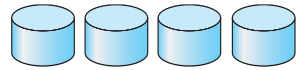
Inverse of probability that the two will fail within 10 hours

Colorado State University

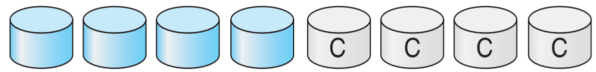
RAID Techniques

- **Striping** uses multiple disks in parallel by splitting data: higher performance, no redundancy (ex. RAID 0)
- **Mirroring** keeps duplicate of each disk: higher reliability (ex. RAID 1)
- **Block parity: One Disk hold** parity block for other disks. A failed disk can be rebuilt using parity. Wear leveling if interleaved (RAID 5, double parity RAID 6).
- Ideas that did not work: Bit or byte level level striping (RAID 2, 3) Bit level Coding theory (RAID 2), dedicated parity disk (RAID 4).
- Nested Combinations:
  - RAID 01: Mirror RAID 0
  - RAID 10: Multiple RAID 1, striping
  - RAID 50: Multiple RAID 5, striping
  - others

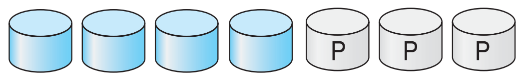
RAID Levels



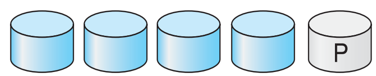
(a) RAID 0: non-redundant striping.



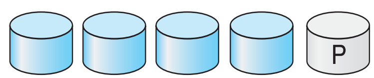
(b) RAID 1: mirrored disks.



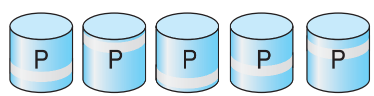
(c) RAID 2: memory-style error-correcting codes.



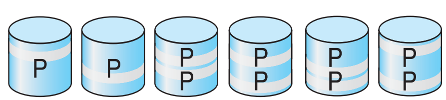
(d) RAID 3: bit-interleaved parity.



(e) RAID 4: block-interleaved parity.



(f) RAID 5: block-interleaved distributed parity.



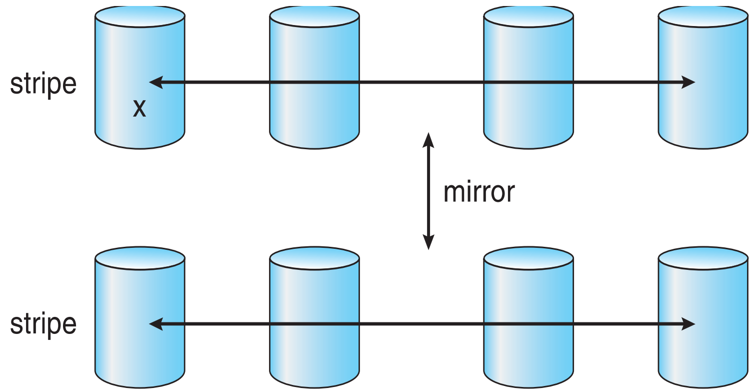
(g) RAID 6: P + Q redundancy.

Not common:
RAID 2, 3,4

Most common
RAID 5

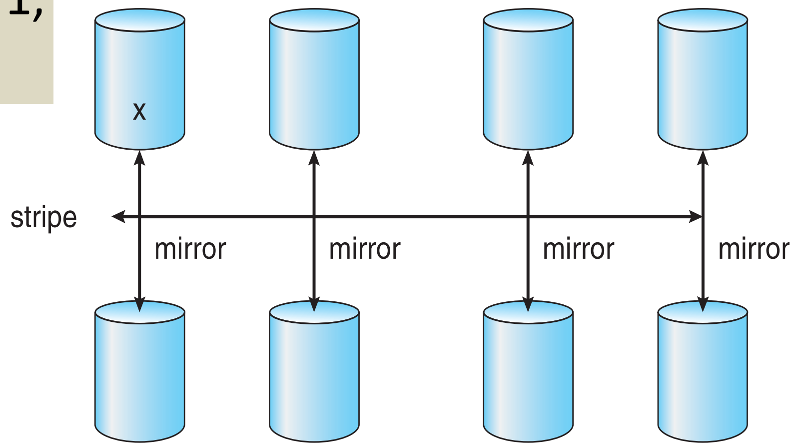
Colorado State University

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



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

RAID 01: Mirror RAID 0
RAID 10: Multiple RAID 1, striping



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