CS343: Operating System

Storage Management, HDD and Disk ARM Scheduling

Lect36 : 6th Nov 2023

Dr. A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

- Mass Storage
  - **Disk Structure**
  - **Disk Arm Scheduling**
  - RAID Structure
- FS Basic
- FS Implementation
- I/O subsystem
- Device Drivers

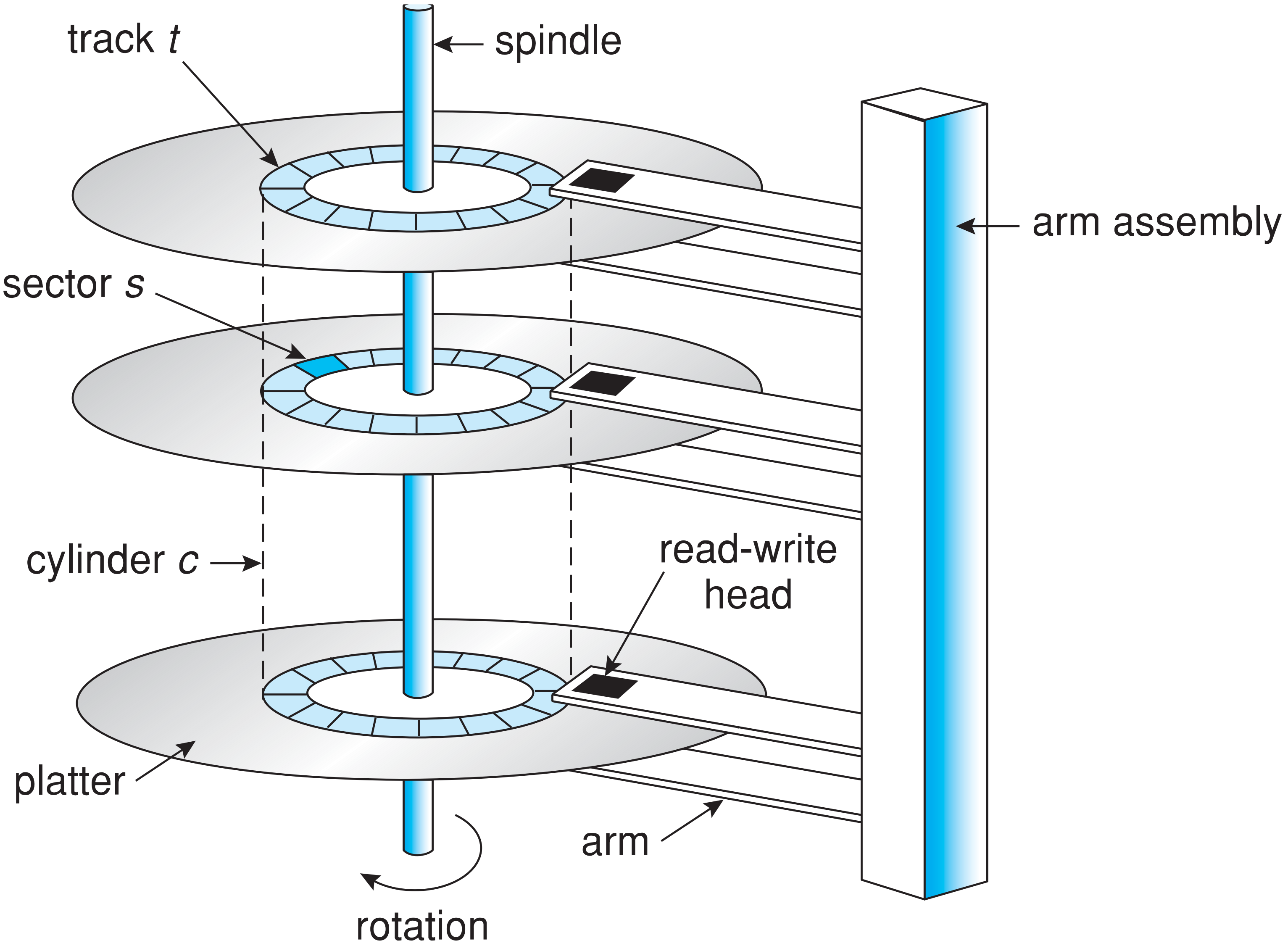
Overview of Mass Storage Structure

- **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 move disk arm to desired cylinder (**seek time**) and 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

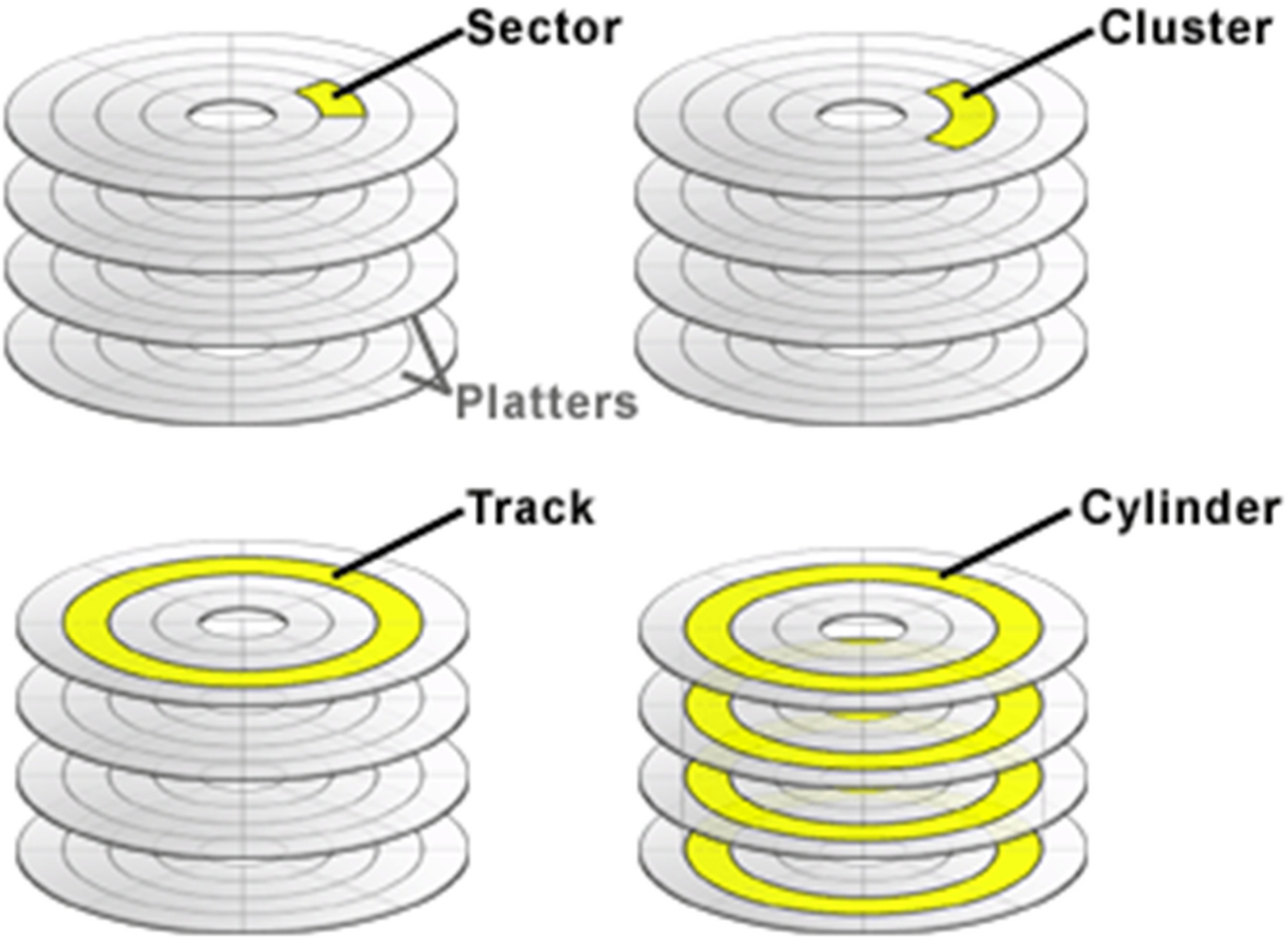
Overview of Mass Storage Structure

- Disks can be removable
- Drive attached to computer via **I/O bus**
  - Busses vary, including
    - **IDE (Integrated Device Electronics)**
    - **Advanced Technology Attachment (ATA)**, **SATA (Serial ATA)**
    - **USB**, **Fibre Channel**, **SCSI (Small Comp Sys Interface), Serial Attached SCSI (SAS)**
  - **Host controller** in computer uses bus to talk to **disk controller** built into drive or storage array

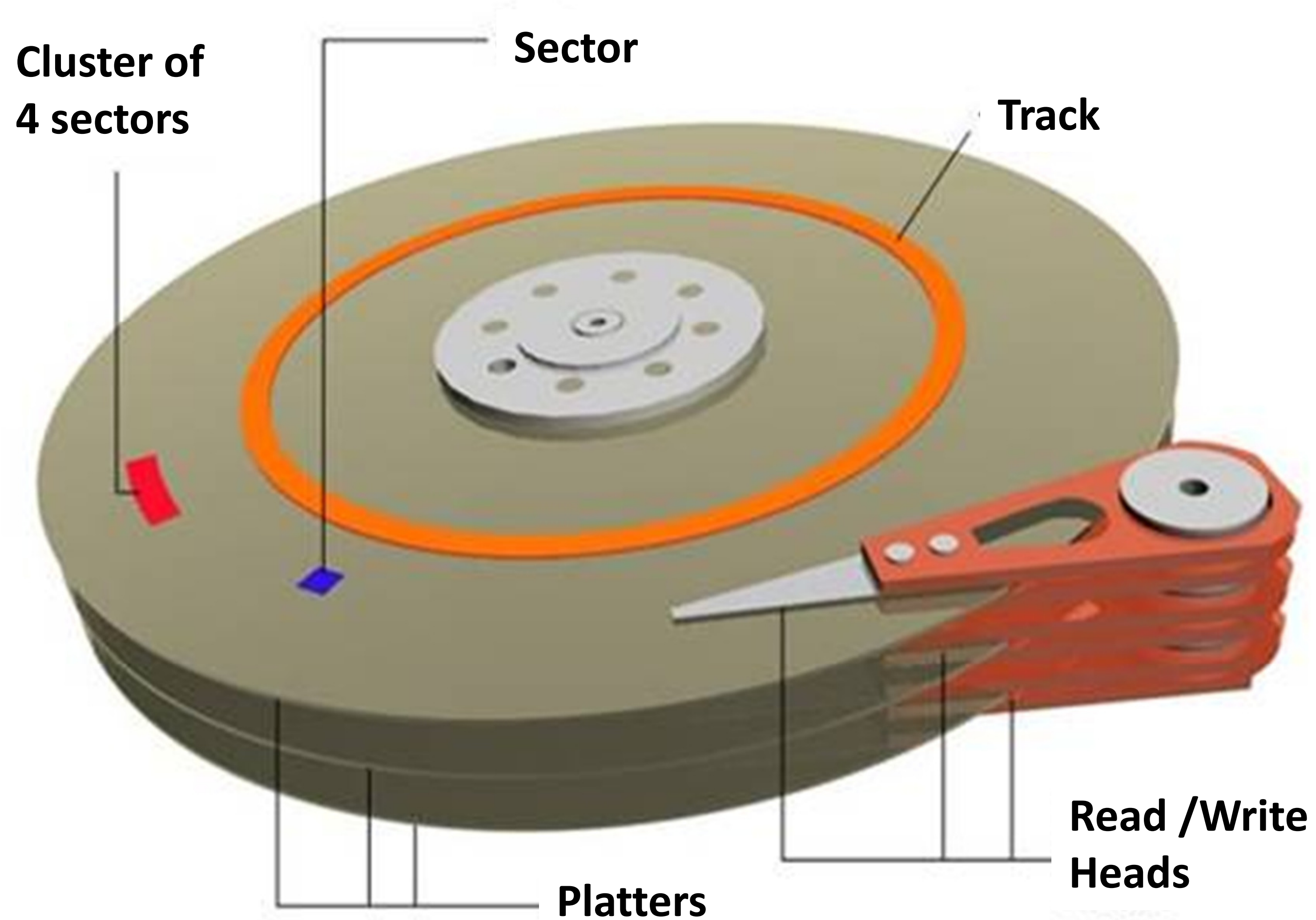
Moving-head Disk Mechanism



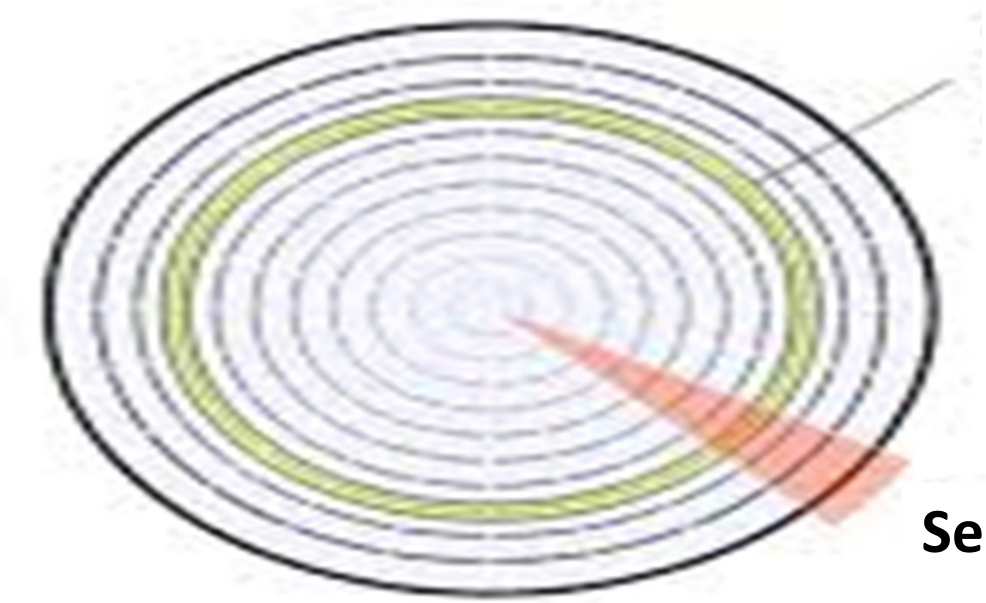
HDD Pictures



HDD Pictures

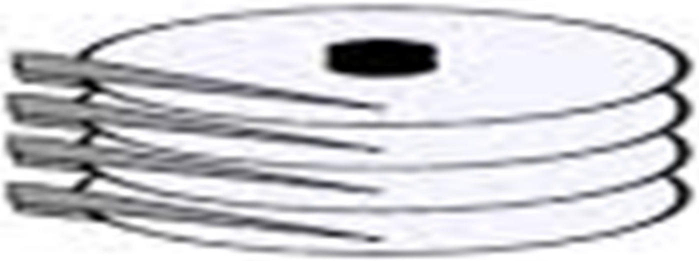


HDD Pictures



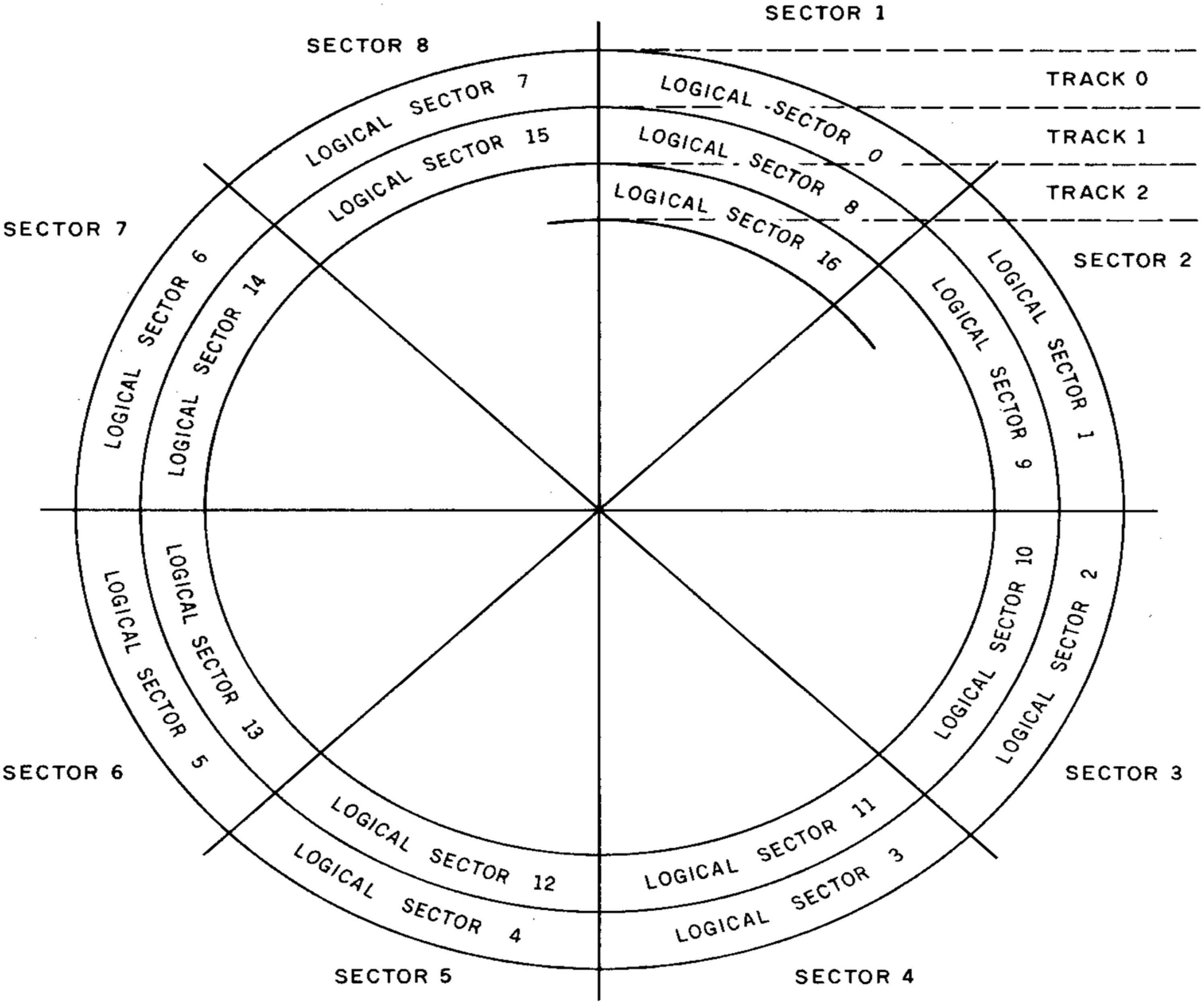
Track/
Cylinder

Sector

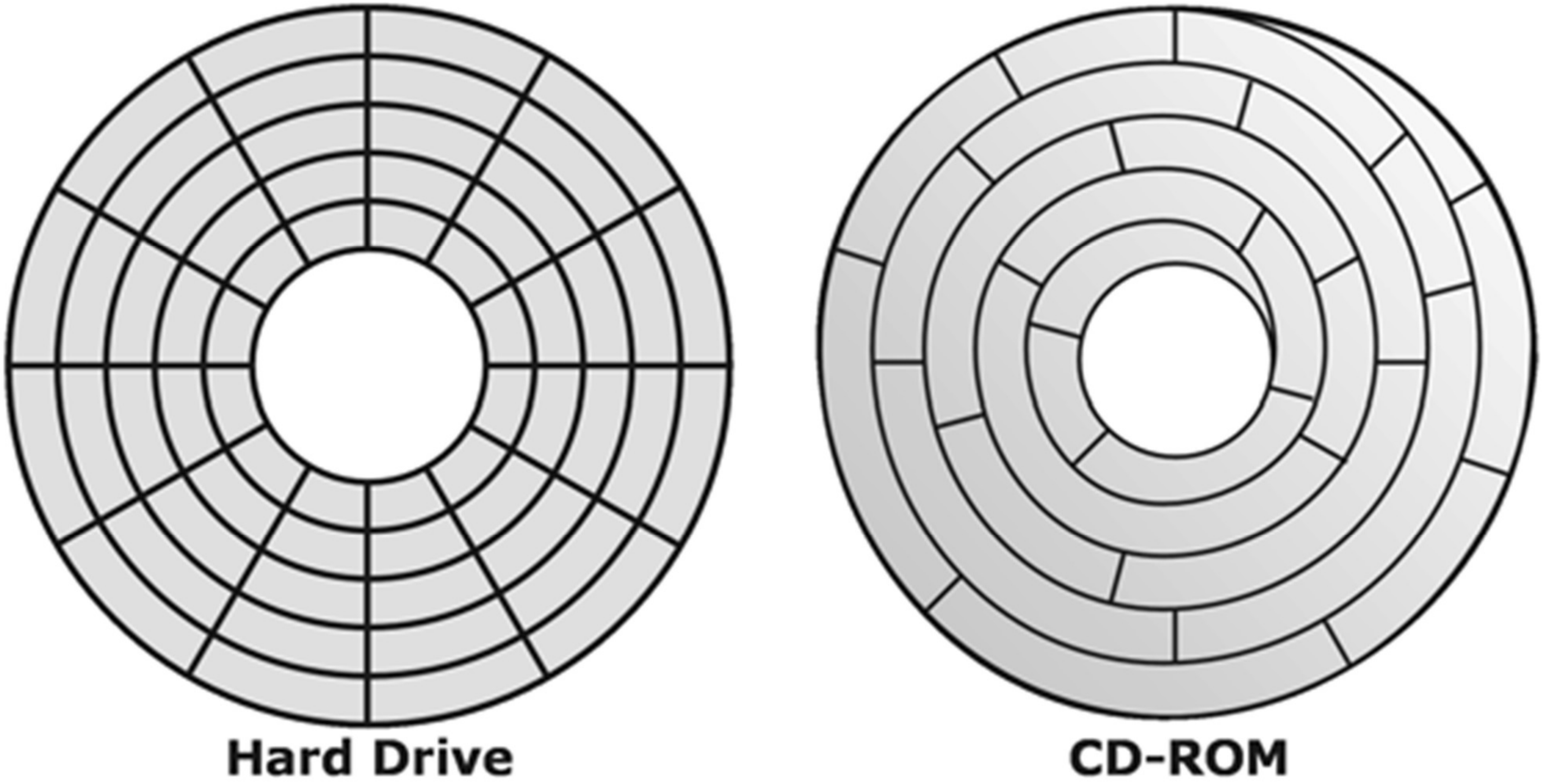


Heads
8 heads
4 platter

HDD Pictures



CD vs HDD



fdisk

```
Disk /dev/hdb: 64 heads, 63 sectors, 621 cylinders
 Units = cylinders of 4032 * 512 bytes
Device       Boot  Start  End   Blocks Id    System
 /dev/hdb    1 * 1 196    395104+     83 Linux
 /dev/hdb2 197     262    133056      82 Linux swap
/dev/hdb3   263    458    395136      83 Linux
/dev/hdb4   459    621    328608      83 Linux
```

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

Hard Disk Performance

- For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a 0.1ms controller overhead =
  - 5ms + 4.17ms *(ctrl ovhd)* + 0.1ms + transfer time =
  - Transfer time = (4KB / 1Gb/s) = 0.031 ms
  - Average I/O time for 4KB block = 9.27ms + .031ms = 9.301ms

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**
- Busses can be too slow -> connect directly to PCI for example
- **No moving parts, so no seek time or rotational latency**

Disk Structure

- Disk drives
  - 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

Disk Structure

- 1-D 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

Disk Scheduling

- OS 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** = Transferred/Time
  - Total number of bytes transferred
  - Total time between first request for service and completion of 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.)

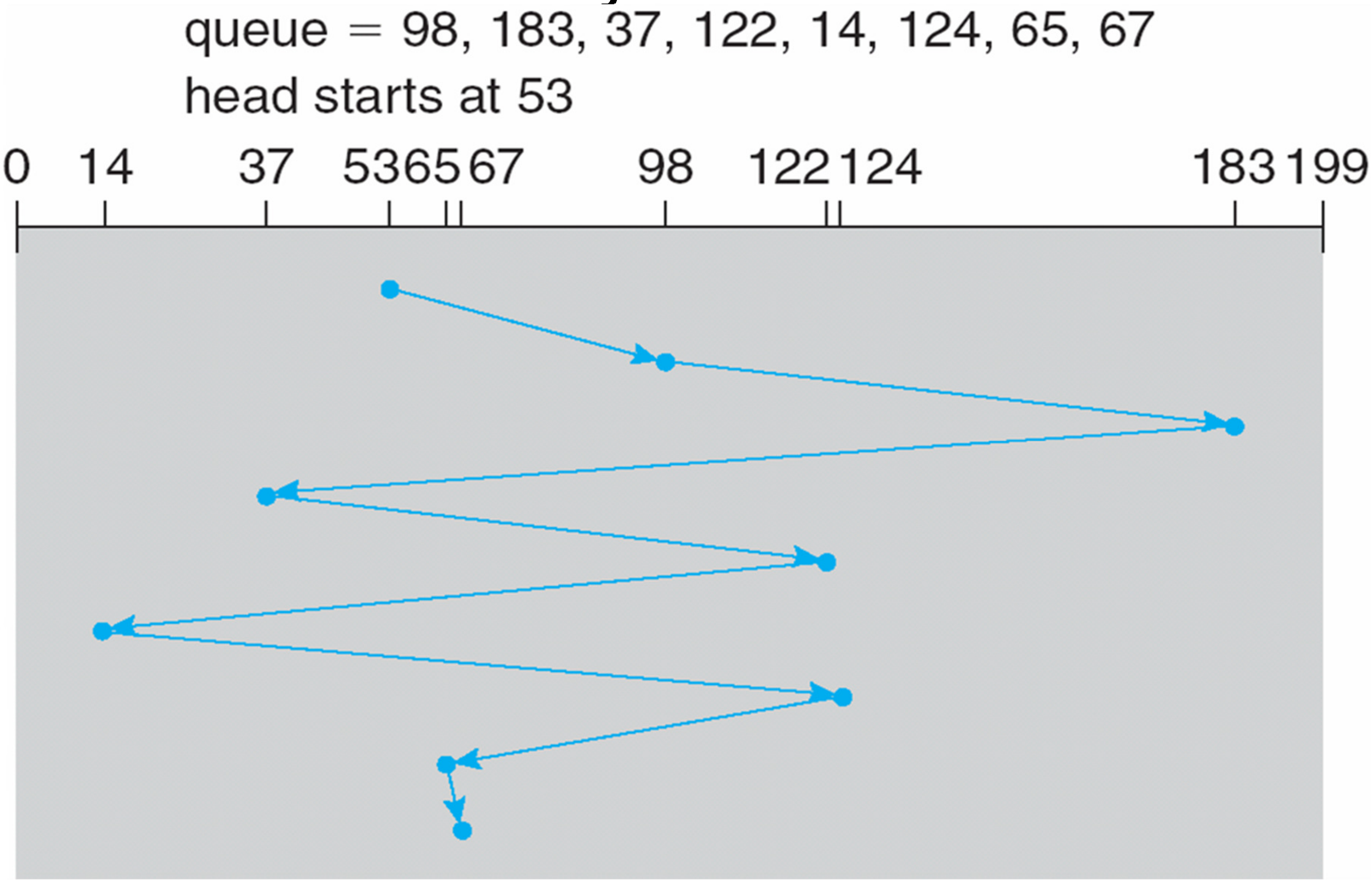
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying “depth”)
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199)

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

Head pointer 53

FCFS

Illustration shows total head movement of **640** cylinders



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

head starts at 53

SSTF

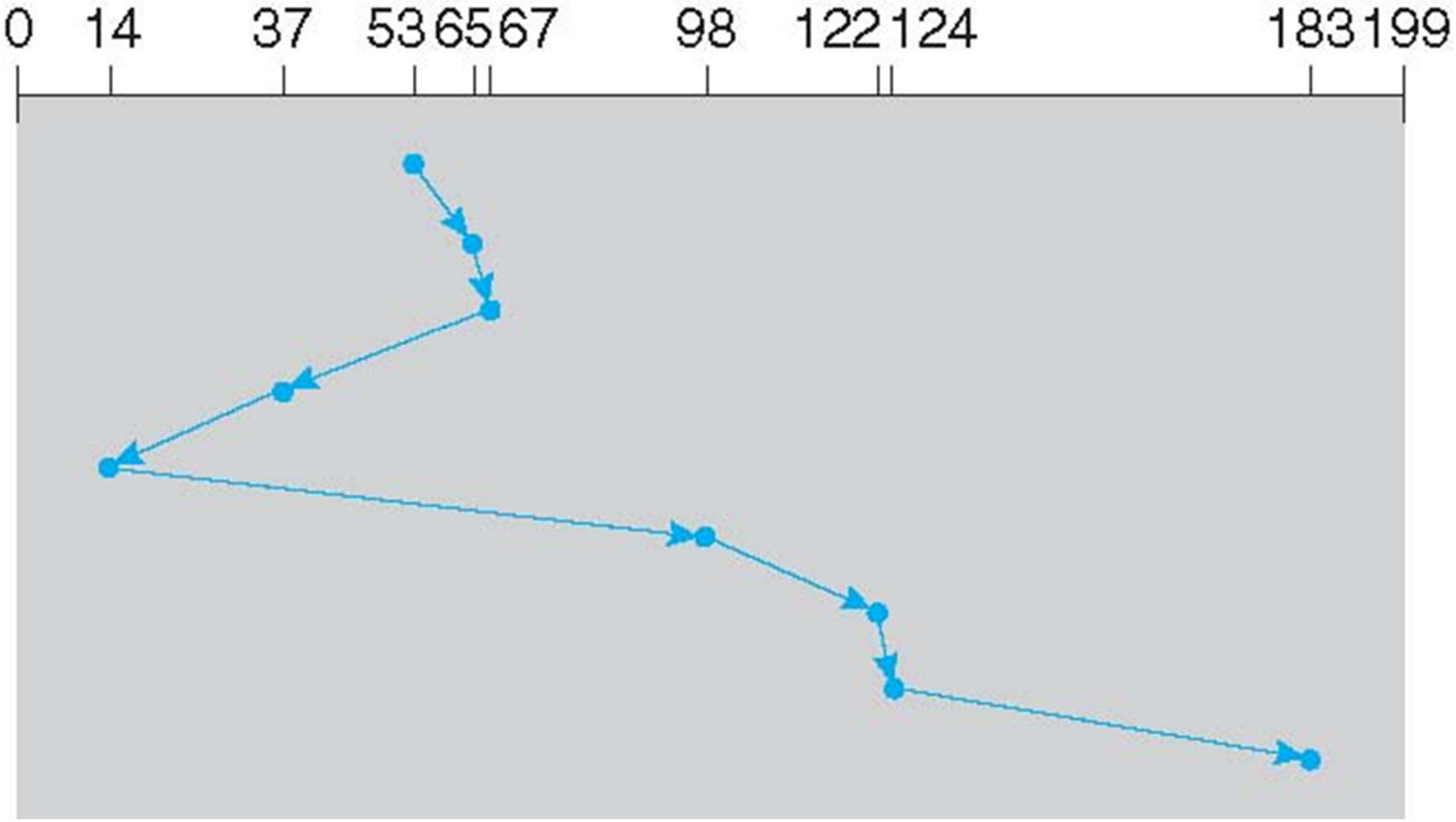
- Shortest Seek Time First selects the request with the minimum seek time from the current head position
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests

SSTF

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

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

head starts at 53

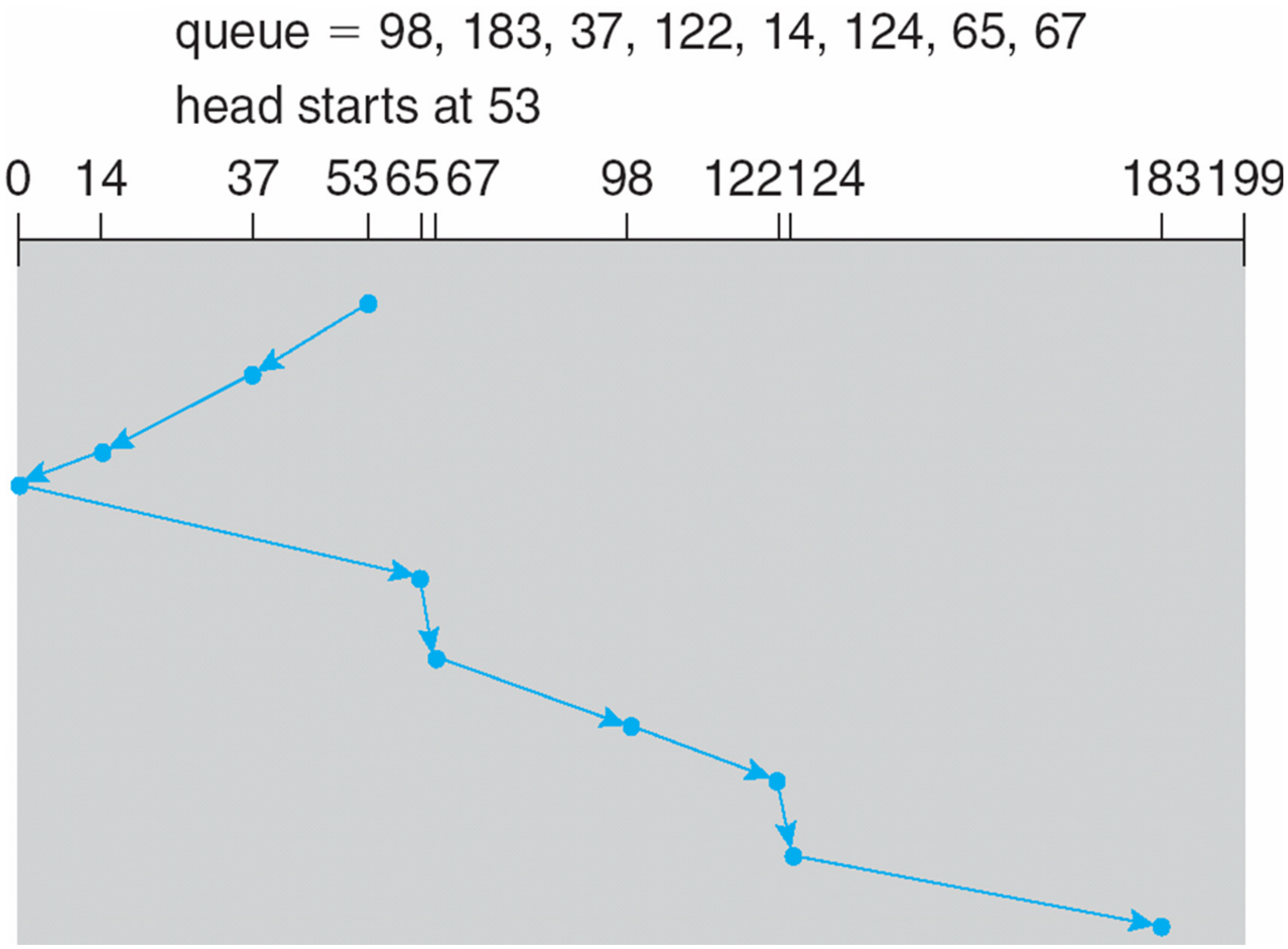


SCAN

- SCAN
  - 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
  - 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 head movement of 208 cylinders



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

head starts at 53

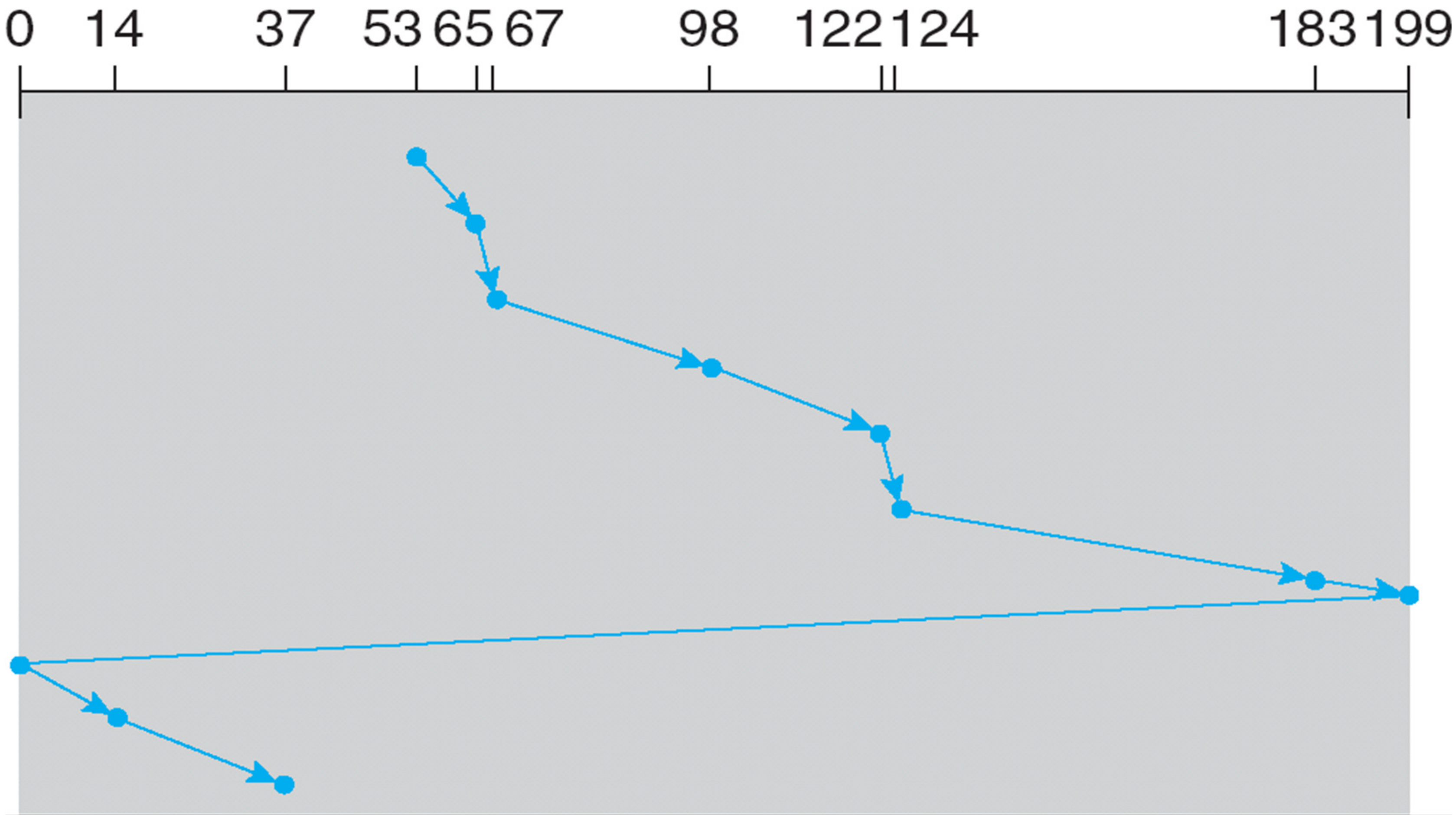
C-SCAN

- 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?
        - Increased…Additional 200

C-SCAN (Cont.)

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

head starts at 53



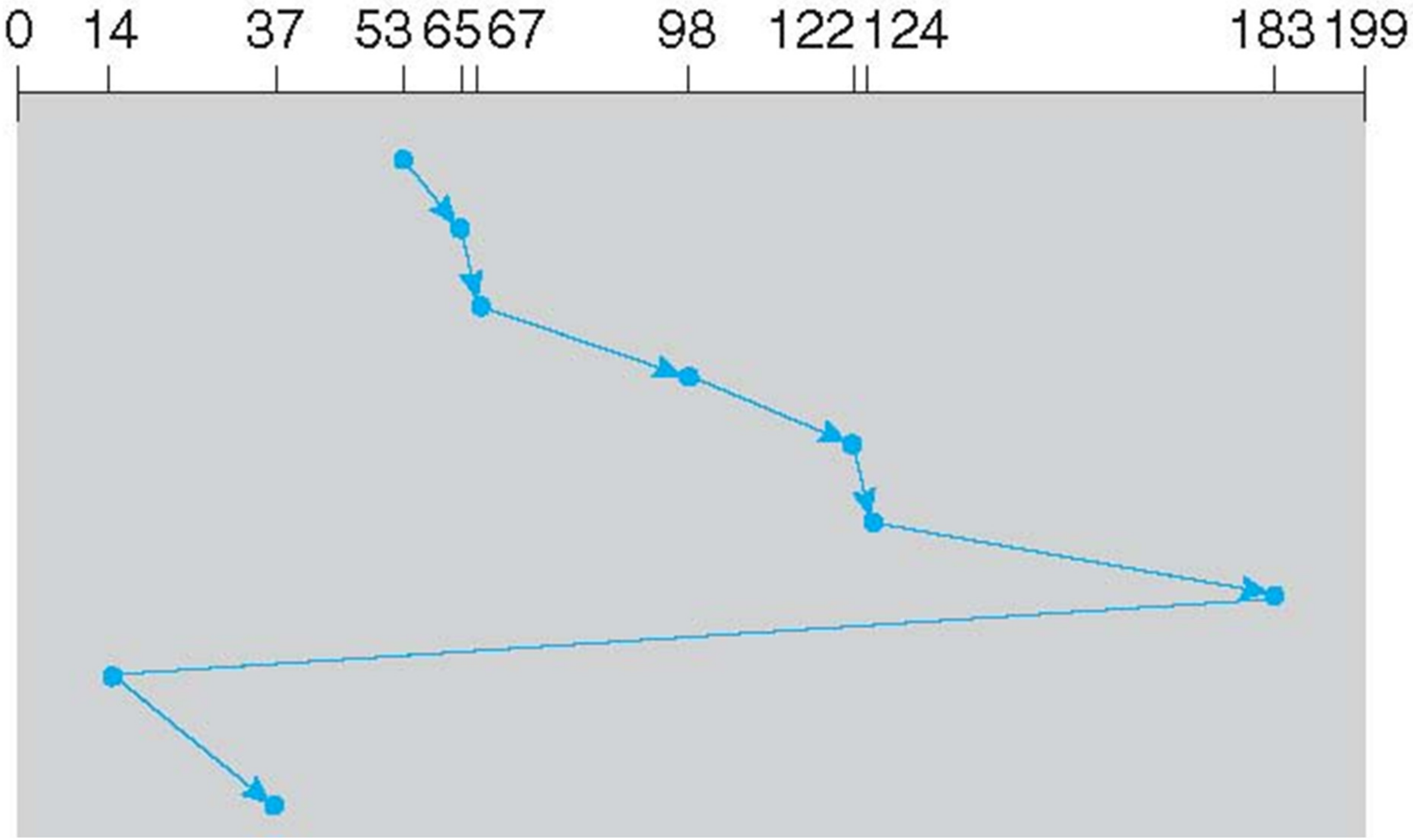
C-LOOK

- 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

C-LOOK (Cont.)

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

head starts at 53



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 number and types of requests

Selecting a Disk-Scheduling Algorithm

- Requests for disk service can be influenced by the file-allocation method :And metadata layout
- Disk-scheduling algorithm should be written as a separate module of the OS, allowing change if necessary
- Either SSTF or LOOK is a reasonable choice for default
- What about rotational latency?
  - Difficult for OS to calculate