L31- DISK SCHEDULING

Disk Storage Systems Management

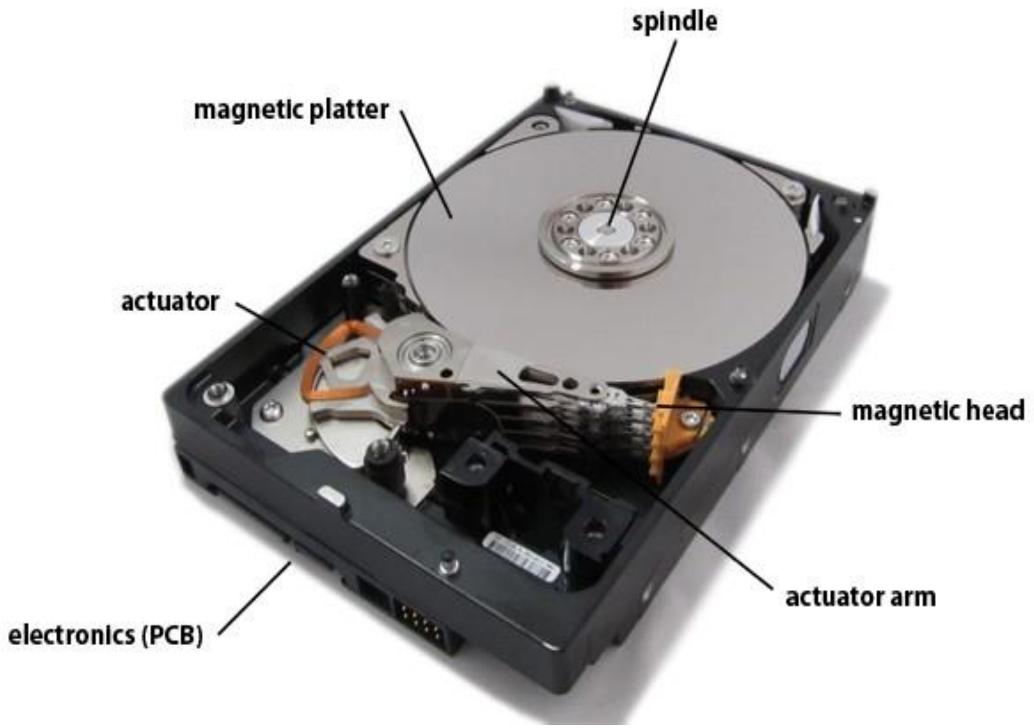
- Disk Organization & Structure
- Disk Attachment
- Disk Scheduling
- Disk Management
- Swap-Space Management
- RAID Structure
- Stable-Storage Implementation

Objectives

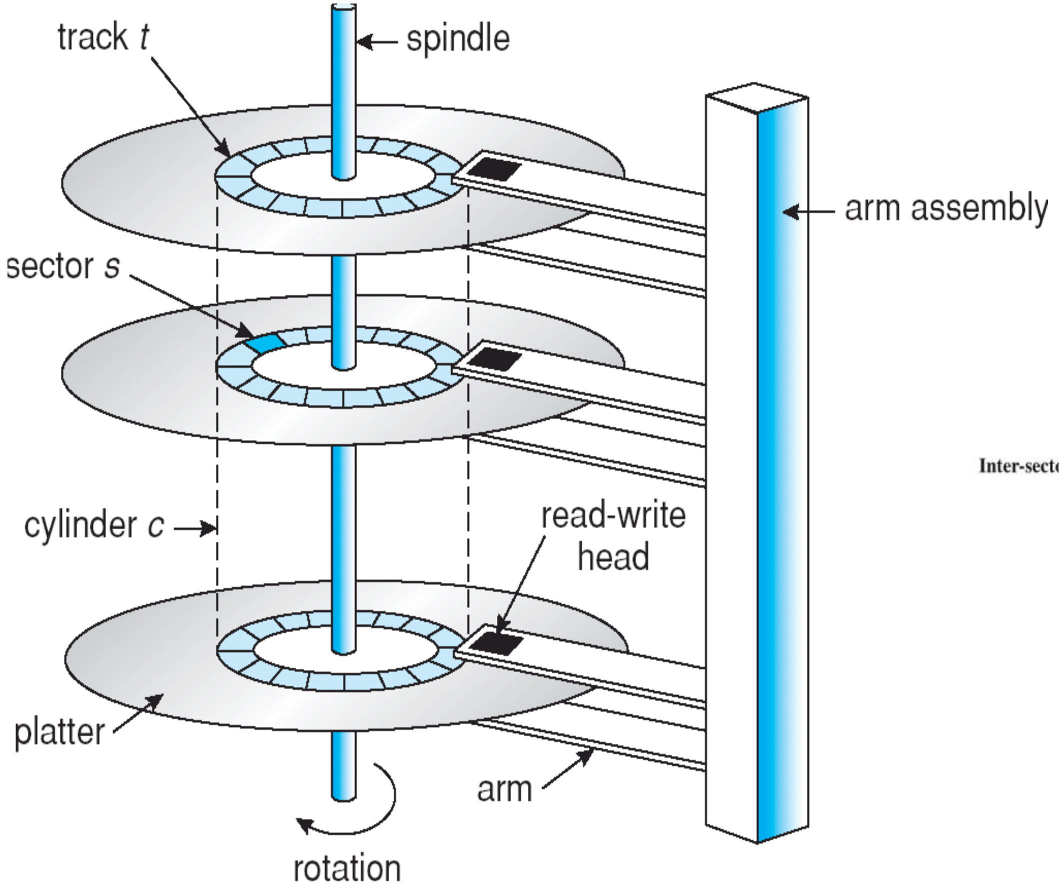
- To describe the physical structure of secondary storage devices and its effects on the uses of the devices
- To explain the performance characteristics of mass-storage devices
- To evaluate disk scheduling algorithms
- To discuss operating-system services provided for mass storage, including RAID

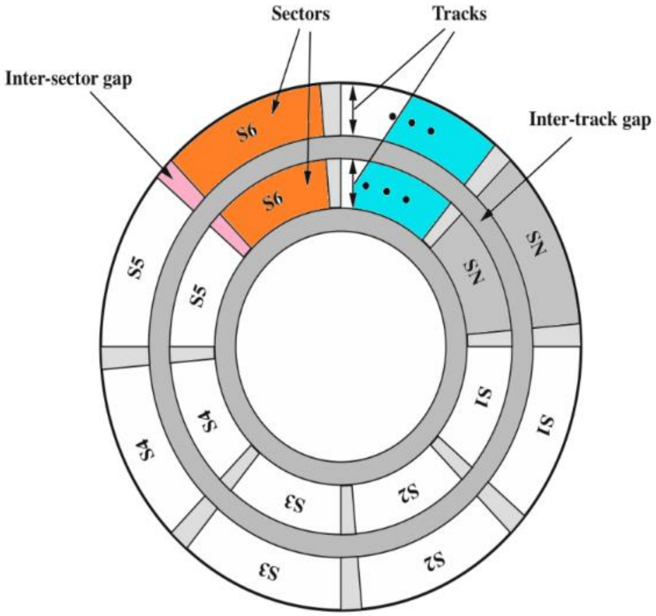
Mass Storage - Hard Disk Drive

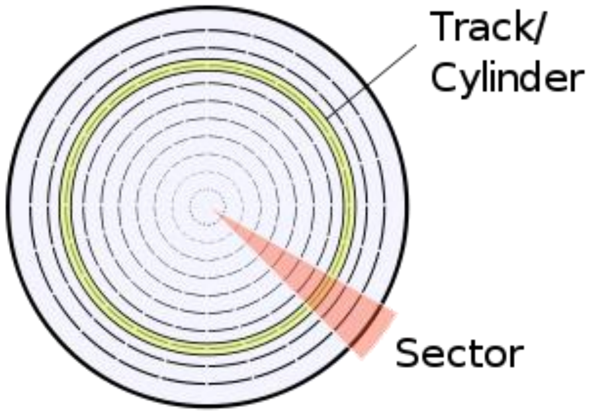
- Systems today need to store many terabytes of data.
- Primary level of permanent storage is hard disk.
- Electromechanical
  - Rotating disks
  - Arm assembly
- Electronics
  - Disk controller
  - Cache
  - Interface controller



Hard Disk Drive Organization

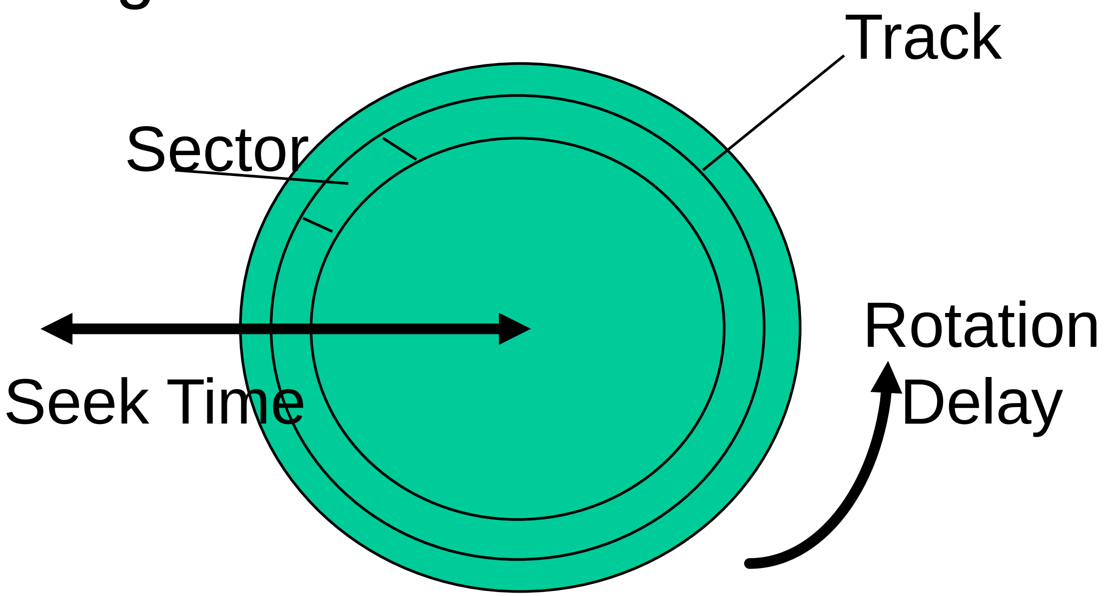






Disk Access Time

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



Disk Scheduling

- ❖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
- ❖ Disk bandwidth is 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

- 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

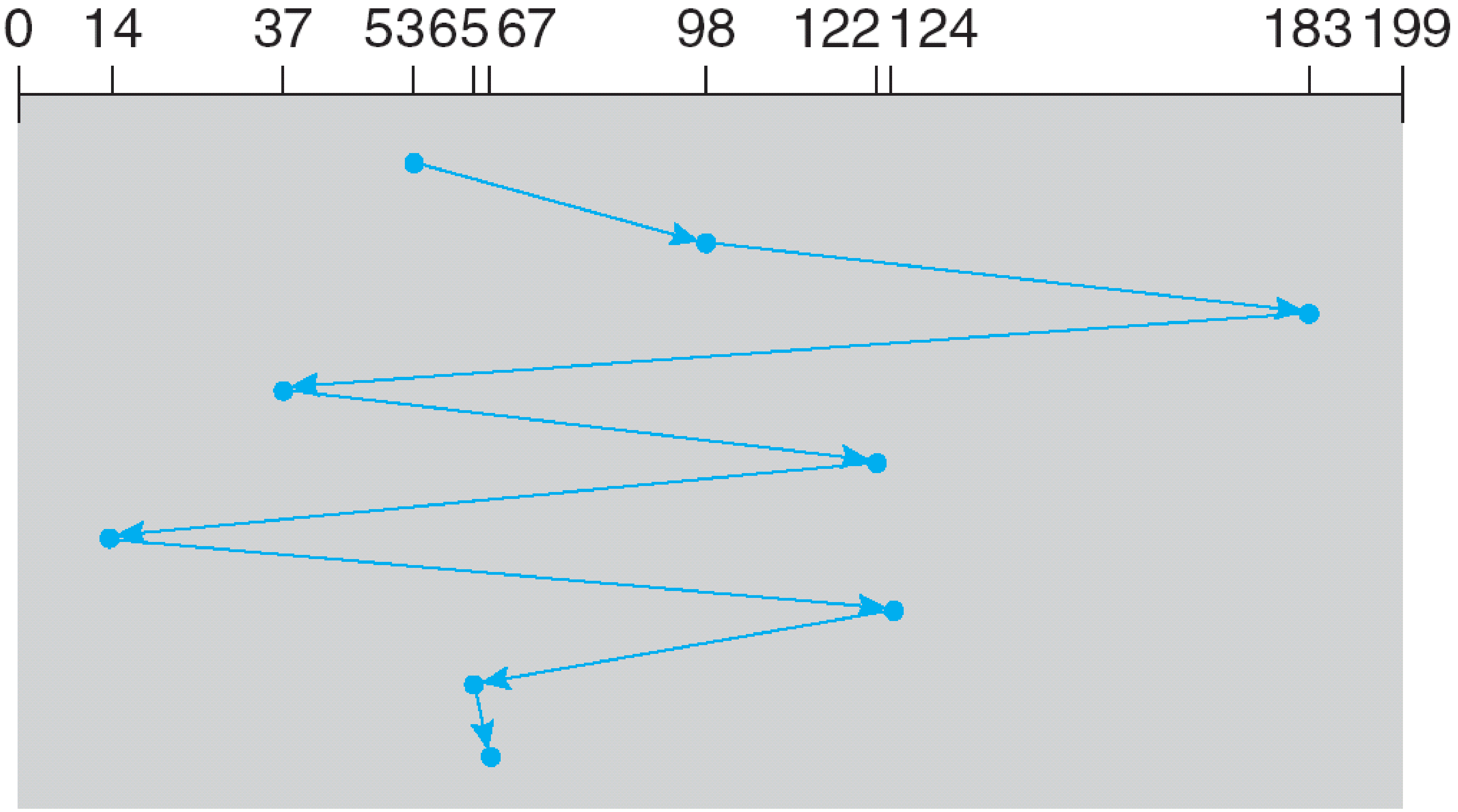
- Drive controllers have small buffers to manage a queue of I/O requests.
- Several algorithms exist to schedule the servicing of disk I/O requests
- Disk Scheduling - The order in which disk cylinder request are serviced so as to optimize average seek time.
  - **FCFS**
  - **SSTF**
  - **SCAN**
  - **C-SCAN**
  - **C-LOOK**
- **Example:** 98, 183, 37, 122, 14, 124, 65, 67 (0-199 cylinders, Head pointer @ 53)

Disk Scheduling Algorithm: FCFS

Total head movements = 640

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

head starts at 53



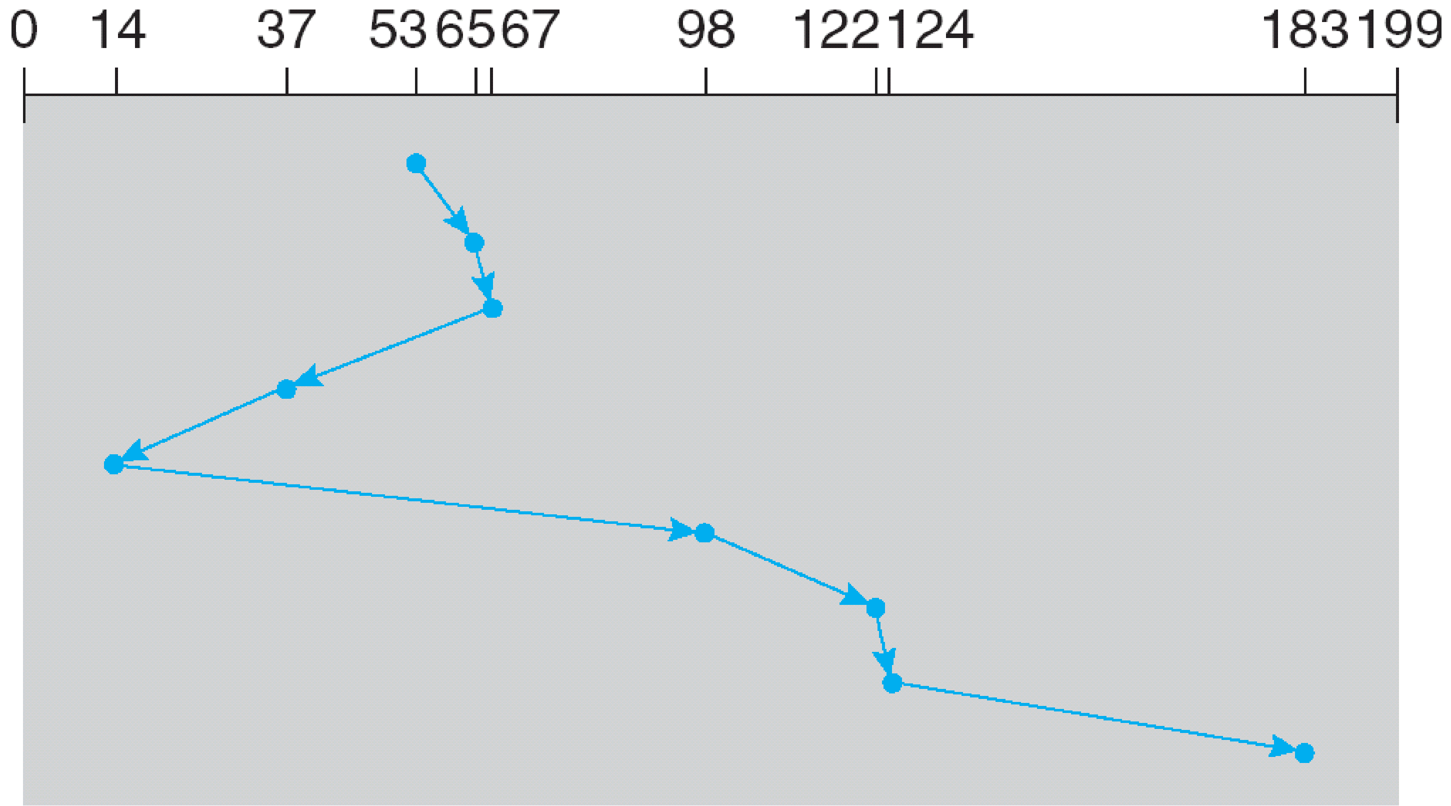
Disk Scheduling Algorithm: SSTF

- Selects request with minimum seek time from current head position,
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests

Total head movement of 236 cylinders

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

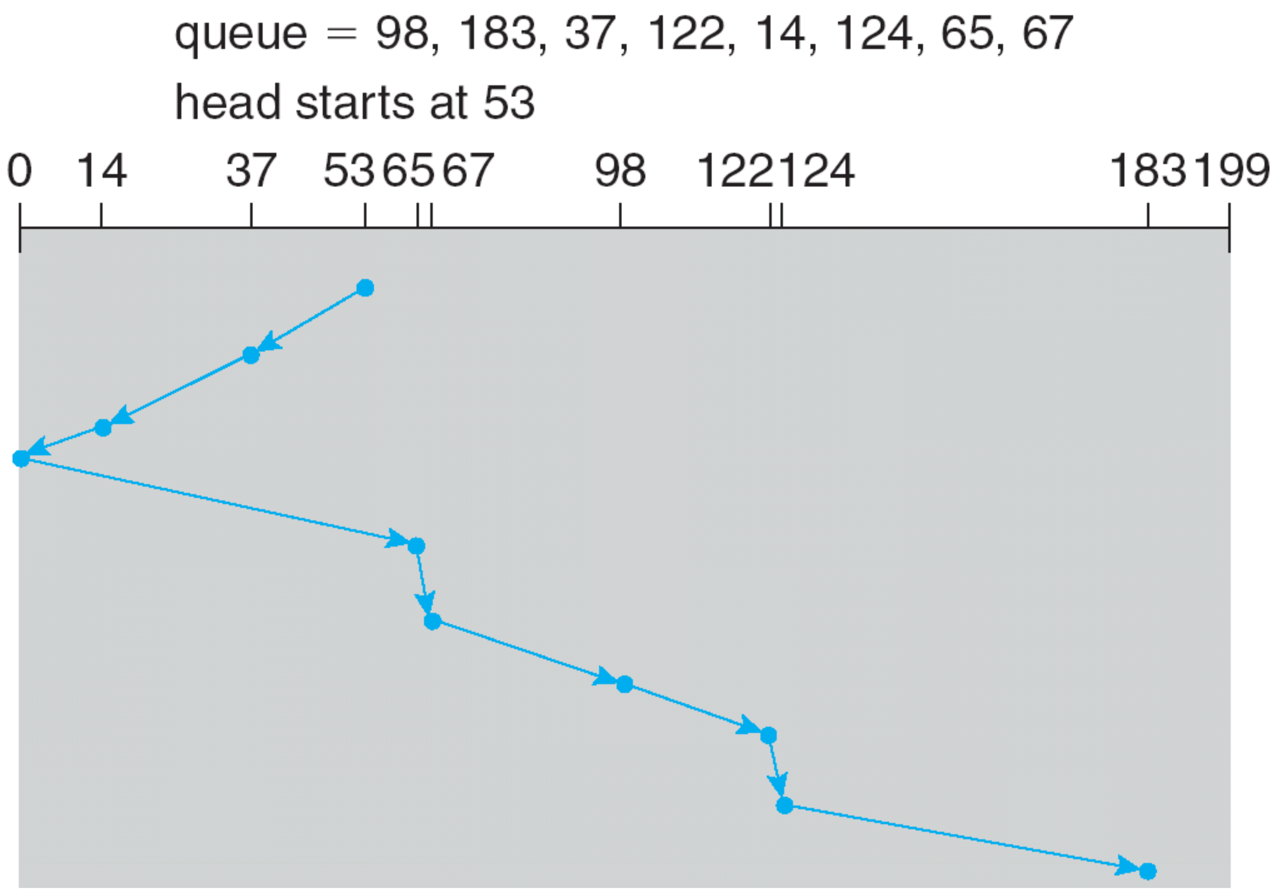
head starts at 53



Disk Scheduling Algorithm: SCAN

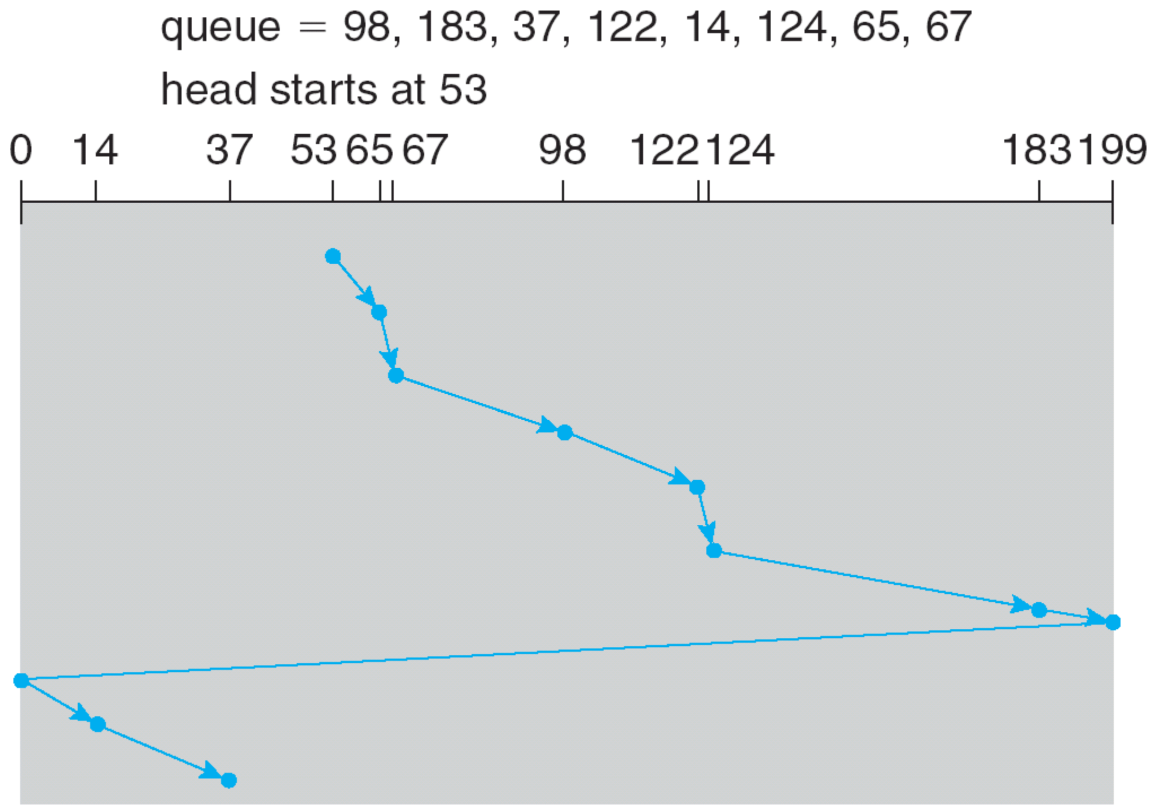
- ❖ The disk arm moves toward one end servicing requests
- ❖ Head movement is reversed when it reach the end and servicing continues. [Also known as elevator algorithm]

Total head movement of 208 cylinders



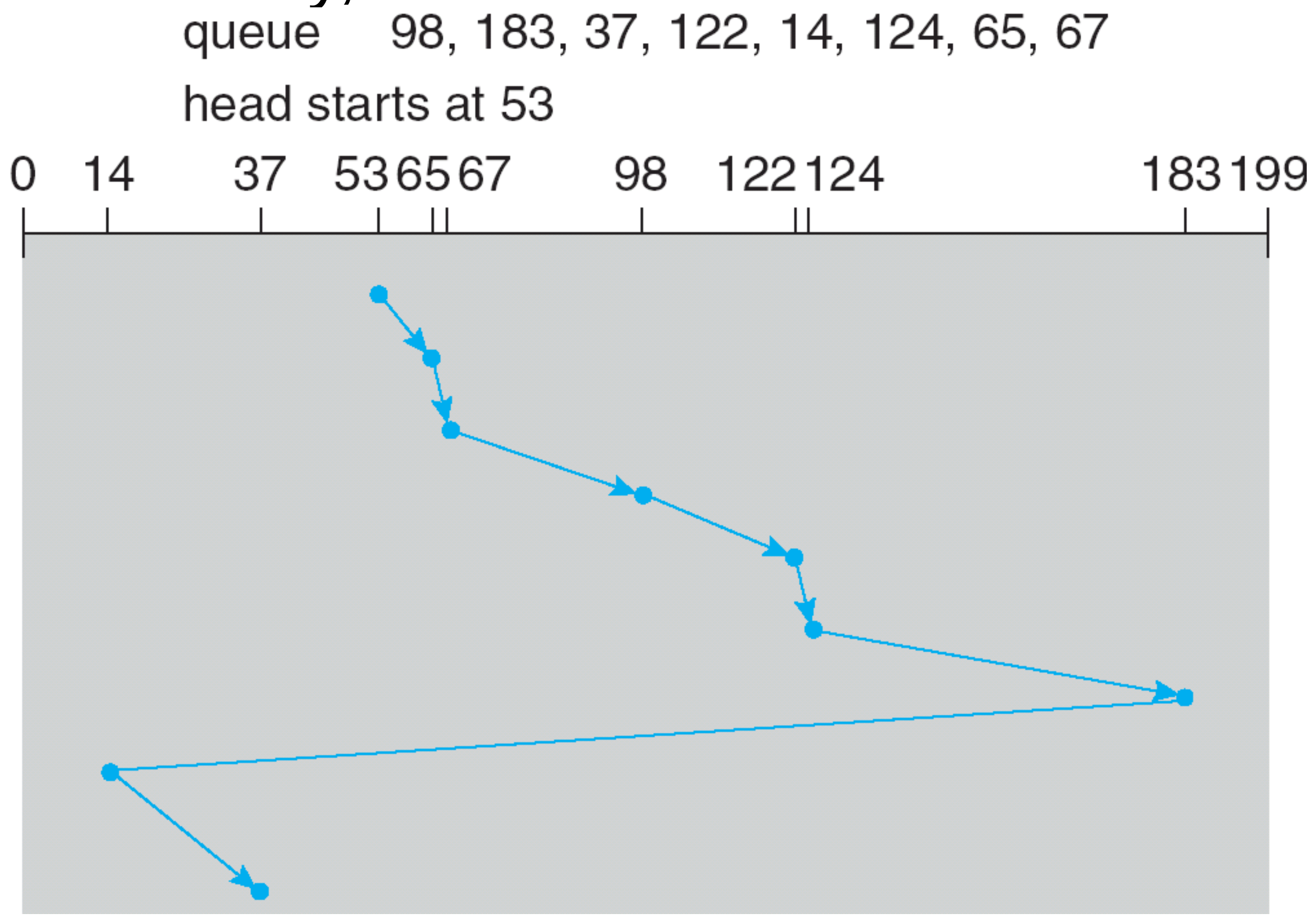
Disk Scheduling Algorithm : C-SCAN

- The head moves from one end of the disk to the other and service the requests as it goes.
- When it reaches the other end it immediately returns to beginning of the disk, No servicing on the return trip.



Disk Scheduling Algorithm : C-LOOK

- Version of C-SCAN
- Arm only goes as far as last request in each direction, then reverses direction immediately,

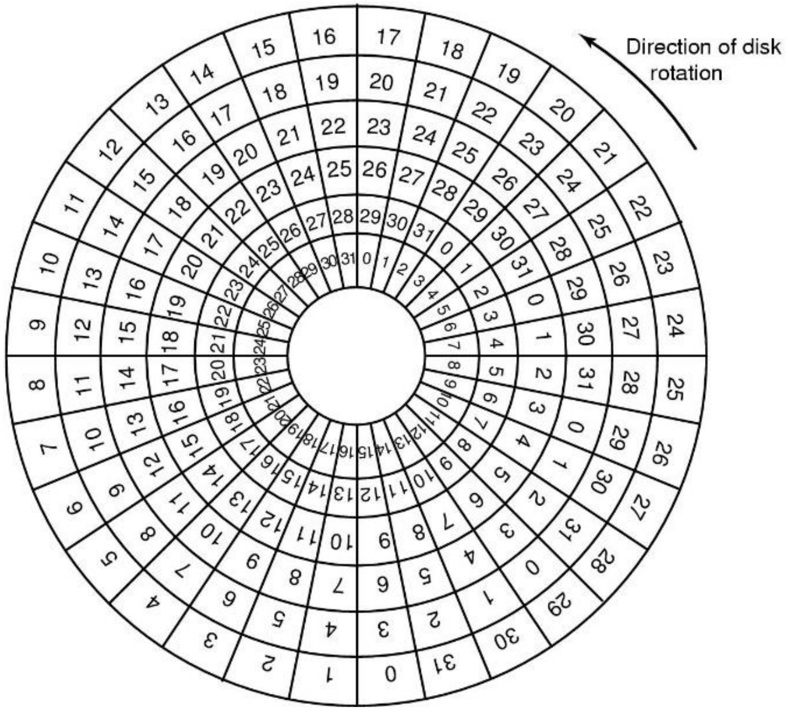


Selecting a Disk-Scheduling Algorithm

- 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

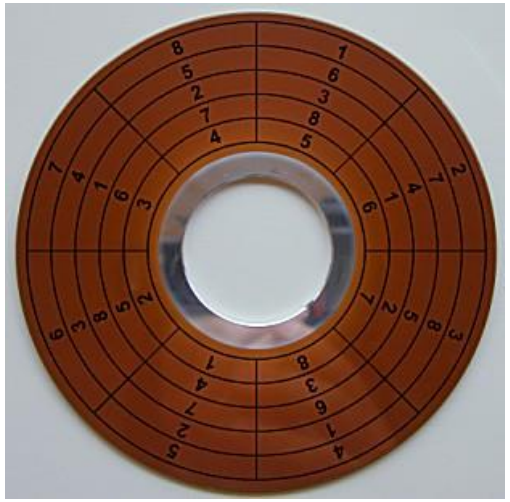
Cylinder Skew

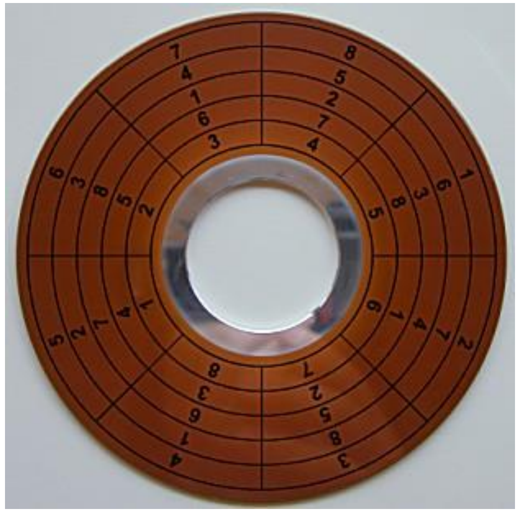
- Why cylinder skew?
- Offsetting the start sector of adjacent tracks to minimize the likely wait time (rotational latency) when switching tracks
- How much skew?
- Example, if 10000 rpm disk drive rotates in 6 ms.
    - Track has 300 sectors
    - New sector every 20 µs
    - If track seek time 800 µs
        - 40 sectors pass on seek
- Cylinder skew: 40 sectors



Head Skew

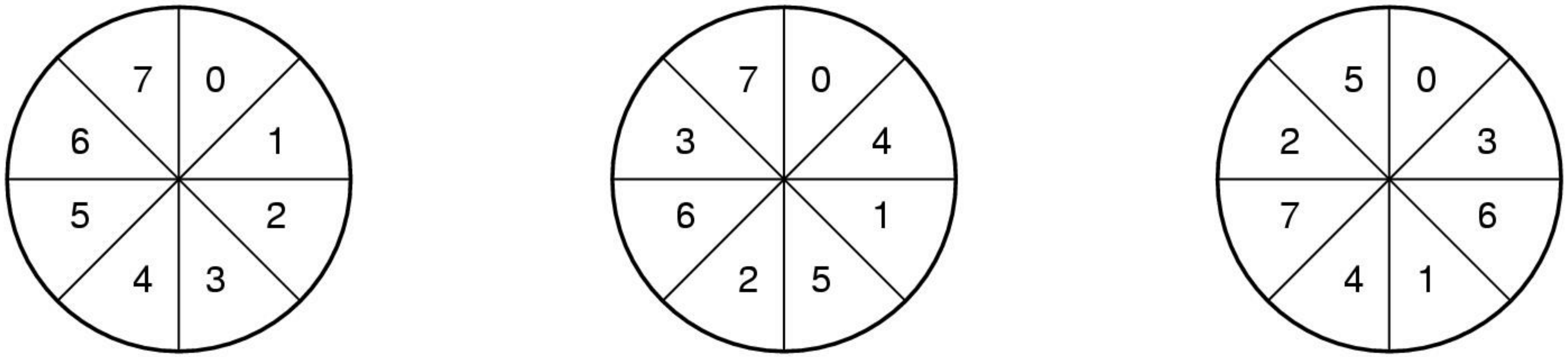
- ❖ Occurs when we change heads within a cylinder, but different platter surfaces.
- ❖ Here there is no physical movement of arm assembly.
- ❖ But it still takes time for the switch from reading one head to reading another.
- ❖ Head skew is the offsetting done on the start sector of tracks of adjacent platters (heads) of same cylinder.





Sector Interleaving

❖ To ensure that sector #n+1 didn't rotate past the head while sector #n was being processed.



(a) No interleaving

(b) Single interleaving

(c) Double interleaving

Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
- Swap-space can be carved out of the normal file system, or, more commonly, it can be in a separate disk partition (raw)
- Swap-space management in various OS.
  - Allocates swap space when process starts; holds text segment (the program) and data segment
  - Uses **swap maps** to track swap-space use
  - Some allocate swap space only when a dirty page is forced out of physical memory, not when the virtual memory page is first created.

Thank You