Operating Systems: Mass Storage Structure

Neerja Mhaskar

Department of Computing and Software, McMaster University, Canada

Acknowledgements: Material based on the textbook Operating Systems Concepts (Chapter 11)

Storage Management

- Main memory too small to accommodate data and programs and is volatile.
- Secondary storage provides additional space for programs to reside permanently.
- An operating system **provides a uniform logical view** of stored information.
- A **file system** provides the mechanism to store/access data on a secondary storage disk (e.g., hard disk drive).

Storage devices

- **Magnetic disks (Hard Disk Drives/HDD)** - provides bulk of secondary storage on modern computers.
- **Solid state disks (SSD)** – Non-volatile efficient and fast memory used like a hard drive.
  - More reliable than HDDs – as no moving parts.
  - Faster than HDDs - as no seek time or rotational latency.
  - However, more expensive and have less capacity.

Storage devices

- **Magnetic Tapes** - Mainly used for backup and storage of infrequently-used data.
  - Holds very large quantities of data
  - Access time slow and Random access ~1000 times slower than HDD.

Magnetic disks

Disk platters that look like CDs

Both sides coated with magnetic material, and information is stored by recording it magnetically on the platter.

Data on a disk drive is read by a **Read write head**. Typically, one head per surface on a separate arm.

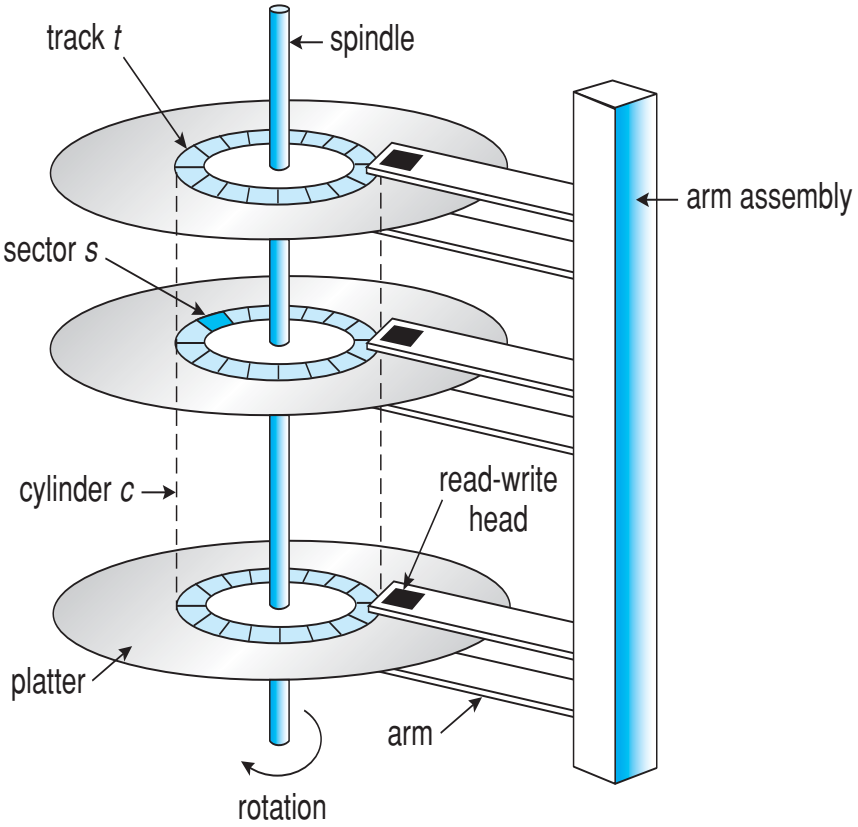
Disks divided into concentric rings called **tracks**

Tracks divided into **sectors**

Read write heads controlled by a common **disk arm** or **arm assembly**, which moves all heads as a unit.

The set of tracks that are at one arm position makes up a **cylinder**

Modern HDDs have 1000s of cylinder and 100s of sectors in each track.



3.5-inch Hard disk drive with cover removed.

Magnetic Disks in Use

- When disk is in use, the drive motor spins/rotates it at high speed (60 to 250 times/sec)
- **Transfer rate** - is rate at which data flows between hard disk drive and computer.
- **Access time** (**positioning time**) consists of two parts:
  - **Seek time –** is the time required for the disk arm to move the heads to the cylinder containing the desired sector.
  - **Rotational latency –** is the time required for the disk to rotate the desired sector to the disk head.
- Seek time and Rotational latency in disks is several milliseconds.
- Several megabytes of data transferred per second

Magnetic Disks Cont…

- **Head crash** results from disk head making contact with the disk surface => Disk damage.
- Disks can be removable
- A disk drive is attached to the CPU and other parts via **I/O bus**
- Host controller in computer uses **I/O bus** to talk to disk controller built into the disk drive or storage array to transfer data.

Disk Structure

- Disk drives are addressed as a large 1-dimensional arrays of **logical blocks** (which is the smallest unit of transfer).
- Each logical block maps to a physical sector (these days of size 4KB).
- Using this mapping a logical block number is converted into a disk address consisting of:
  - a cylinder number,
  - a track number within that cylinder, and
  - a sector number within that track.

Disk Structure

Mapping of logical blocks:

- 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.

Disk Scheduling

- **Disk bandwidth** = Total no. of bytes transferred/ Total time between the first request for service and the completion of the last transfer.
- The operating system is responsible for providing fast access time and large disk bandwidth.
  - Goal of OS is to minimize seek time
- Managing the order in which disk I/O requests are serviced can improve both the access time and disk bandwidth.
  - When I/O requests arrive, they are placed in a queue and serviced based on the disk scheduling algorithms.

Disk Scheduling Continued

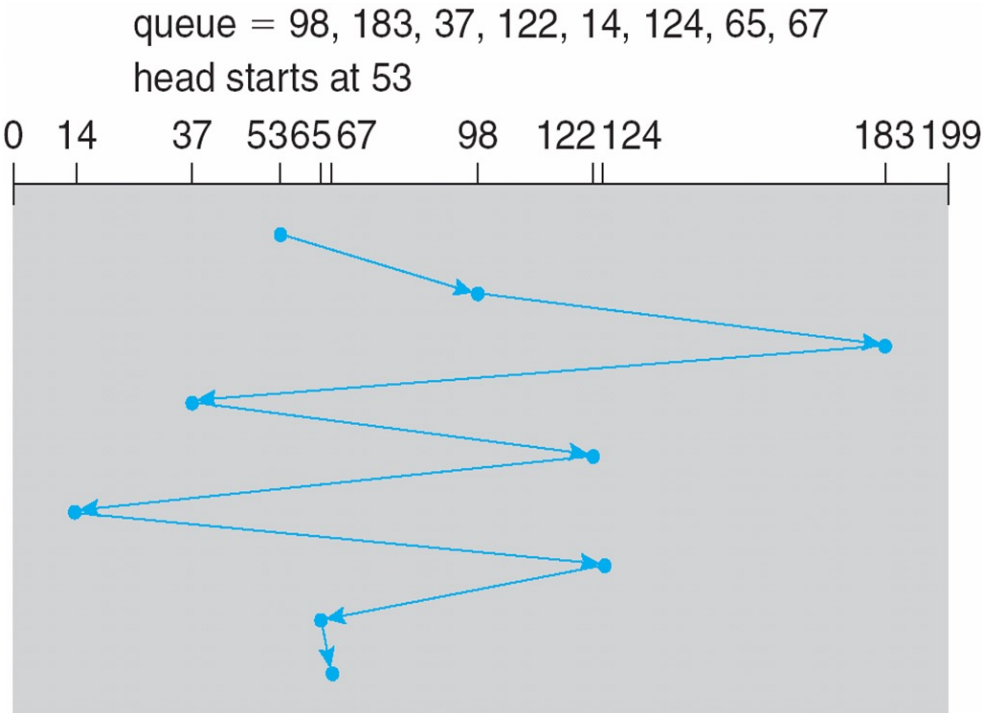
- Several algorithms exist to schedule the servicing of disk I/O requests.
  - First Come First Served - FCFS
  - Shortest Seek Time First - SSTF
  - SCAN
  - C-SCAN
  - LOOK
  - C-LOOK
- These scheduling algorithms are illustrated with the following request queue consisting of cylinder numbers (0-199):

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

- Head pointer at cylinder 53

FCFS

- I/O requests scheduled on first-come first-served basis.
  - Simplest and fair
- Below illustration shows total head movement = = |53-98| + |98-183| + |183-37| + |37-122| + |122-14| + |14-124| + |124-65| + |65-67| = **640 cylinders**.

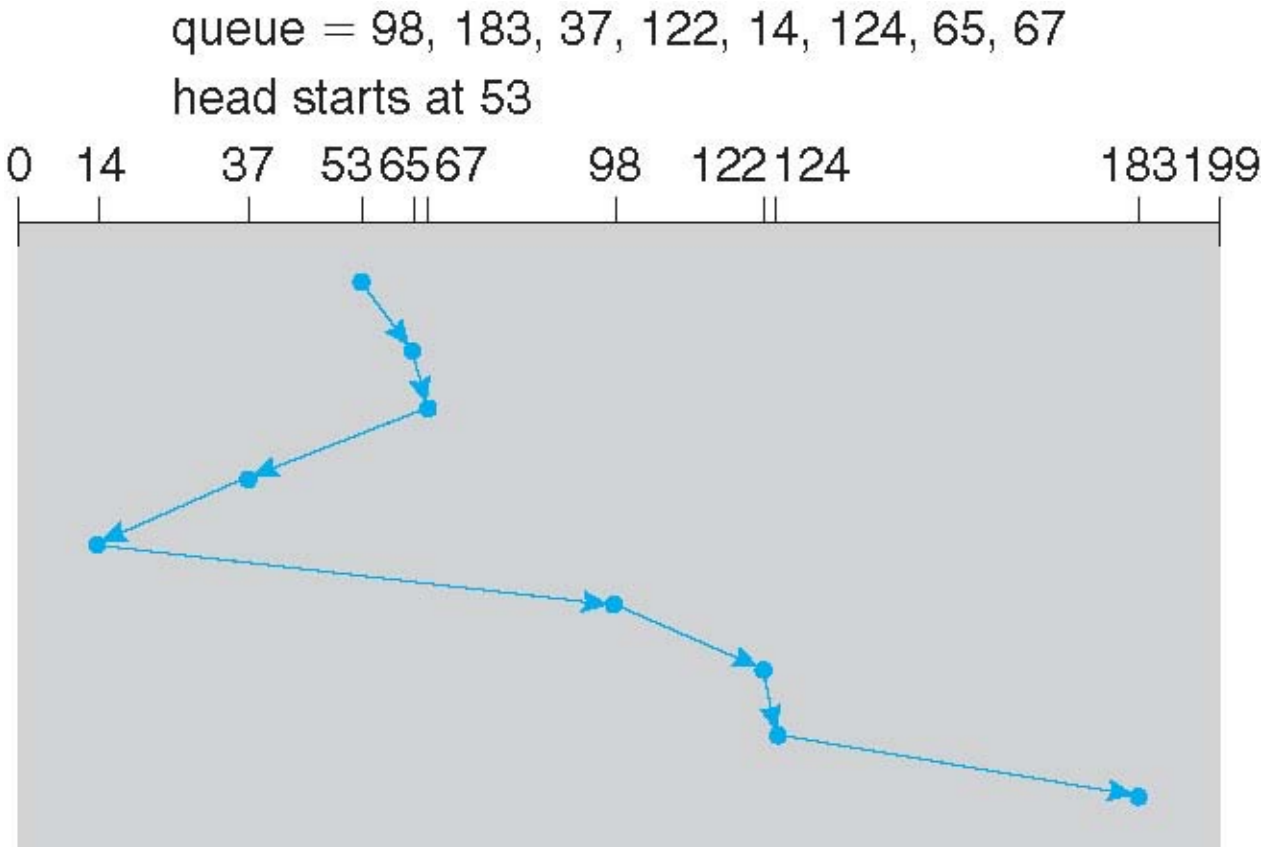


SSTF

- Shortest Seek Time First selects the request with the minimum seek time from the current head position
- In other words, SSTF chooses *the pending request closest to the current head position*.
- SSTF scheduling is a form of SJF scheduling
  - May cause **starvation of some requests**
- SSTF is most common

SSTF Cont…

- Order the requests: 14, 37, 65, 67, 98, 122, 124, 183.
- Total head movement of = |53-65| + |65-67| + |67-37| + |37-14| + |14-98| + |98-122| + |122-124| + |124-183| = 236 cylinders.

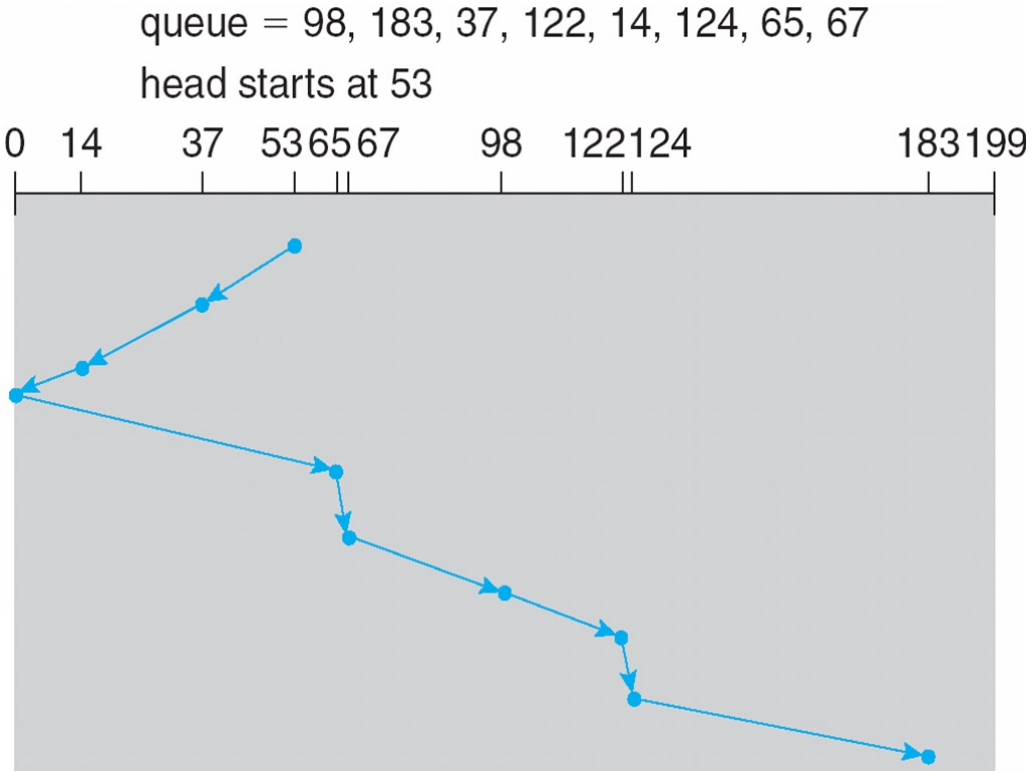


SCAN

- The disk arm moves in the direction of the head, while servicing requests along the way , till it reaches the end of the disk.
- Then it 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.
- The head continuously scans back and forth across the disk – sometimes called **elevator algorithm**
- Problem: Requests waiting at the other end; that is opposite to the head movement wait the longest.

SCAN Cont…

- Head direction is ‘LEFT’.
- Total head movements = |53 - 0| + |0 – 183| = 236

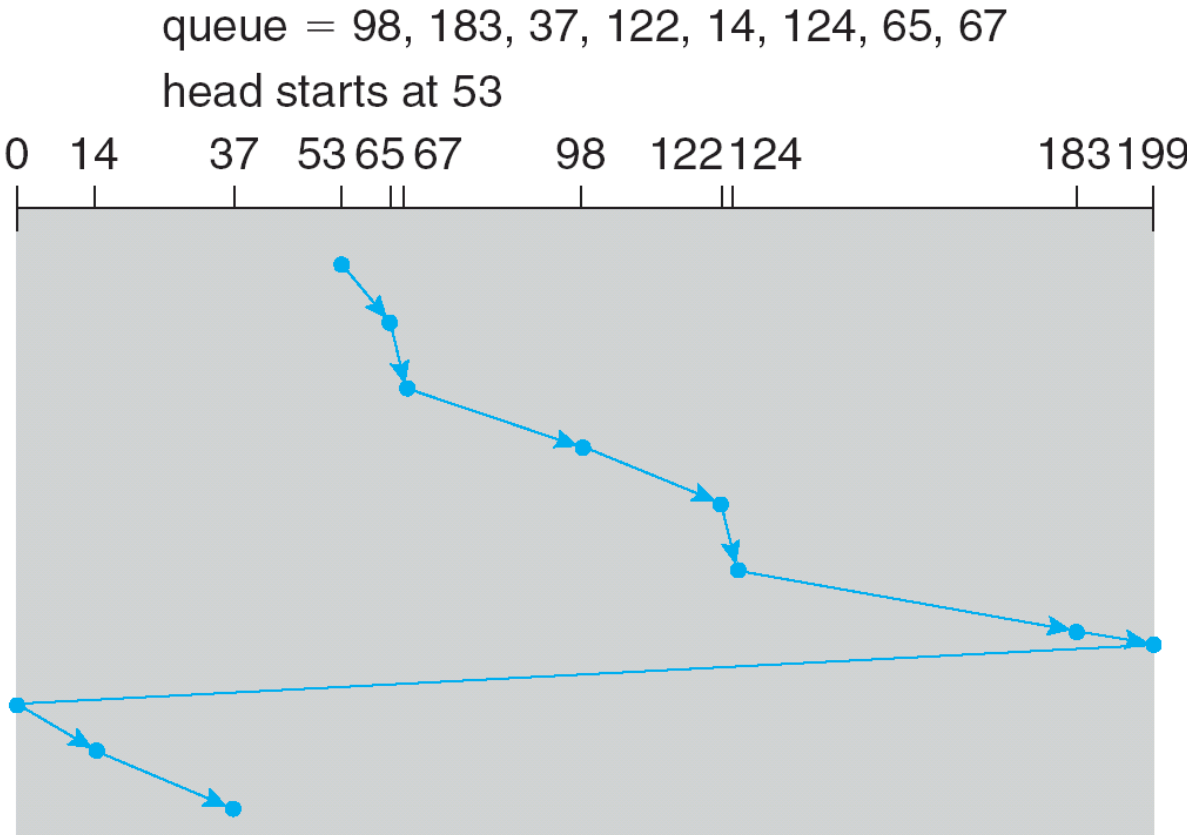


C-SCAN

- C-SCAN algorithm is a variant of SCAN designed to provide 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. We will count the total head movements when the head reverses its direction and goes to the other end of the disk. Therefore, we added I199-0I in next example.
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one

C-SCAN Continued…

- Head direction is ‘RIGHT’.
- Total head movements = |53 - 199| + |199-0| + |0 - 37| = 183 + 199 = 382.

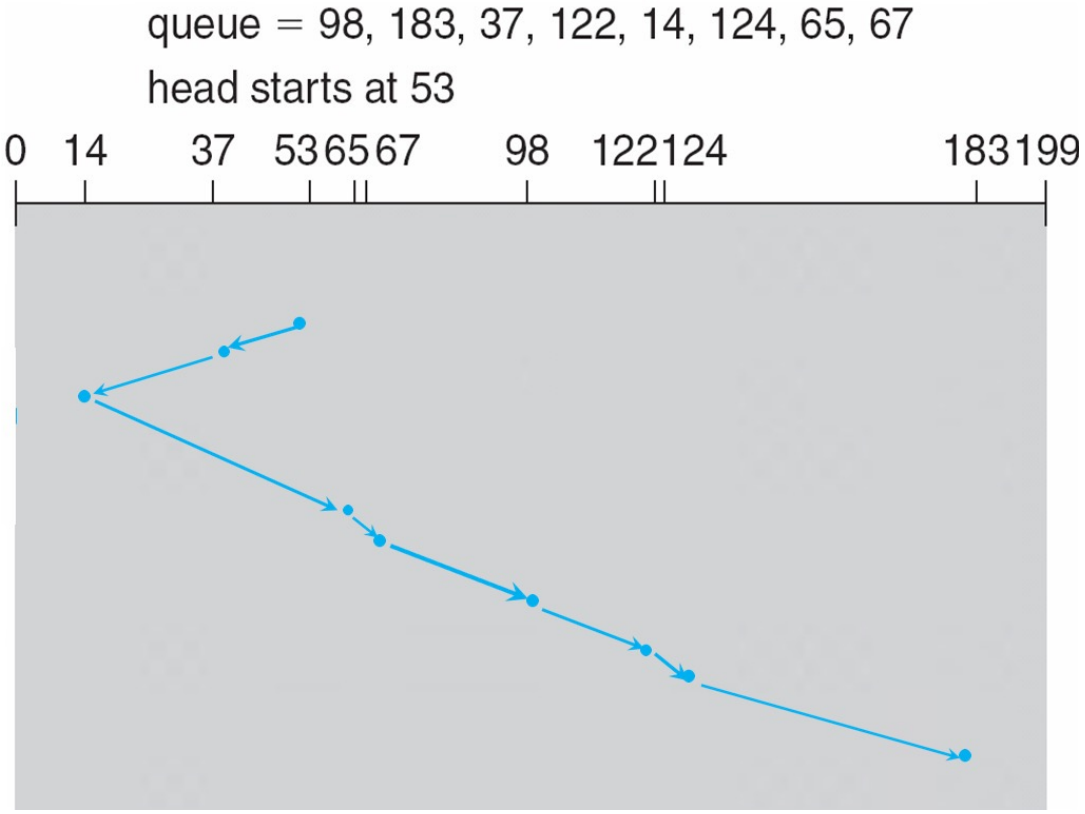


LOOK

- LOOK is a version of SCAN
- The 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.
- LOOK is also another common disk scheduling algorithm

LOOK Example

- The total number of head movements with head direction = LEFT using the LOOK algorithm is = |53 – 14| + |14 – 183| = 208

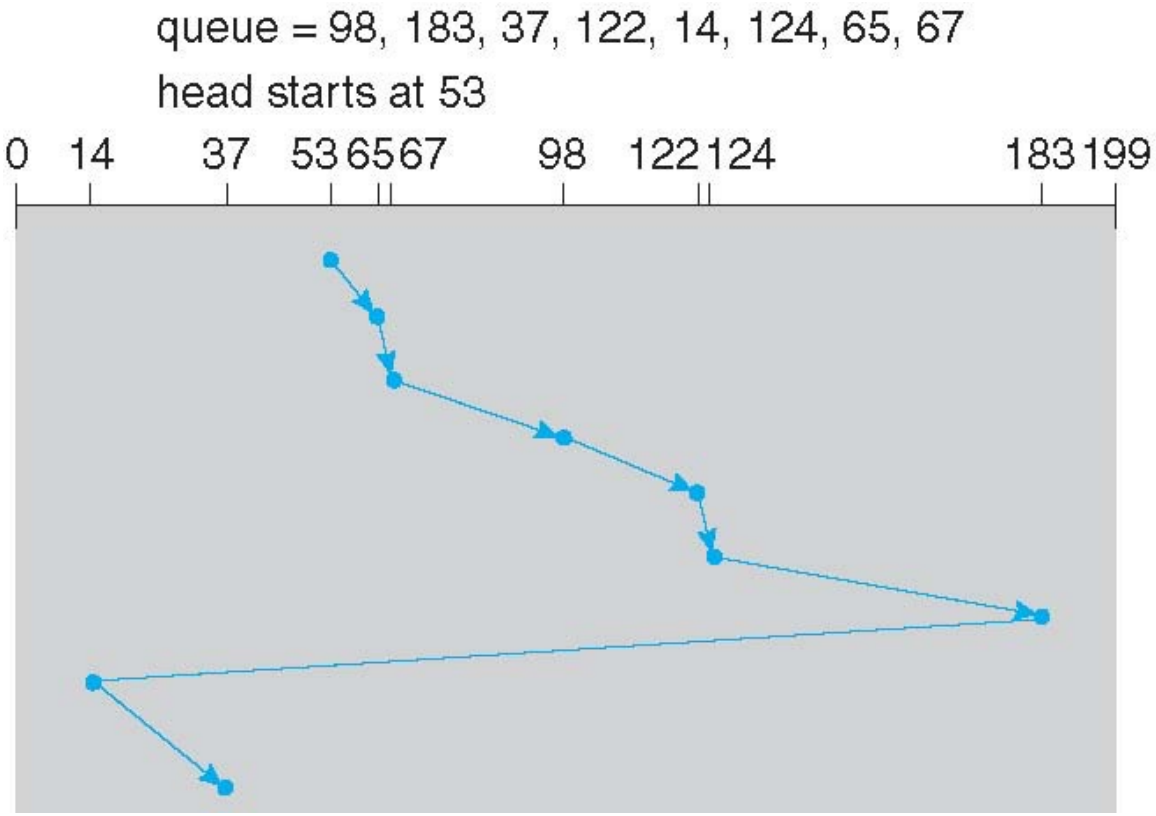


C-LOOK

- C-LOOK a version of C-SCAN
- When the algorithm services the last request in a direction, it immediately reverses its direction and goes to the last request (say a) in this changed direction without servicing any requests.
  - We will count the head movements when the head reverses its direction and goes to the last request in this changed direction. Therefore, we add |183-14| in the next example.
- Before services this request it turns direction and then starts servicing the requests starting from a.

C-LOOK

- Total head movements with head direction = 'RIGHT' using C-LOOK is = l 53 - 183l + ~~l183-199l + l0-14l +~~ l183-14l + l14-37l = 322.
- Here, the request referred to as a in the previous slide is 14.



Disk Management

- The operating system is responsible for several other aspects of disk management apart from disk scheduling. For example:
  - **Disk Initialization**
  - **Booting from disk**
  - **Bad-block recovery**

OS Disk Initialization

- To use the disk, the OS needs to record its own data structures on the disk.

  This is done in two steps:

  - **Partition the disk –** partition the disk into one or more groups of cylinders
    - Each partition is treated as a logical disk.
  - **Logical formatting** or "creating a file system" - the operating system stores the initial file-system data structures onto the disk.
  - **Mounting** - Making a file system available for use by logically attaching it to the root file system.

Disk Formatting

- To use a disk to hold files, the following steps are followed:
  - **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, plus data, plus error correction code (**ECC**)

    - Usually, 4KB of data but can be selectable
    - Initialize the mapping from logical block numbers to defect-free sectors on the disk
    - Done at factory level where disks are made

Boot Block

- Bootstrap program is the initial program to run on a system
  - It is stored in ROM – inflexible (as it is read only)
    - **Bootstrap loader** program stored in ROM instead
      - Full bootstrap program stored in boot blocks at fixed location on disk.
      - A disk that has a boot partition is called a **boot disk** or **system disk**

Bad-block recovery

- Disk controller maintains a list of bad blocks
  - During low level formatting, some sectors are set aside (not visible to OS) as spare sectors.
  - When bad block detected, content restored from backup onto one of the spare sectors.
  - Any reference to the bad sector is forward to the spare sector. This is called **sector sparing or forwarding**

Swap space management

- Swap space can be carved out of the normal file system, or it can be in a separate partition.
- No file system or directory structure is placed in this space, but **swap manager** manages this space.
- Optimized for speed than storage – due to increased number of accesses
- Suffers from internal fragmentation – but this problem is short lived
  - Processes or its pages reside in swap space for shorter duration.
  - Any fragmentation problem is resolved when the system is initialized/reset at boot time.
  - In the past Linux suggested to set up swap space = 2 × physical memory. Now, it is far less.