University Institute of Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Bachelor of Engineering (Computer Science & Engineering)

Subject Name : Operating System

Chapter : Disk Scheduling and Disk Management

Prepared By: Er. Inderjeet Singh(e8822)

DISCOVER . LEARN . EMPOWER

Contents

- Overview of Mass Storage Structure
- Disk Structure
- Disk Attachment
- Disk Scheduling
- Disk Management
- Swap-Space Management
- RAID Structure
- Disk Attachment
- Stable-Storage Implementation
- Tertiary Storage Devices
- Operating System Support
- Performance Issues

Device Management

- Device management in operating system implies the management of the I/O devices such as a keyboard, magnetic tape, disk, printer, microphone, USB ports, scanner, camcorder etc.as well as the supporting units like control channels. The basics of I/O devices can fall into 3 categories:
- **Block device:** it stores information in fixed-size block, each one with its own address. For e.g. disks.
- **Character device:** delivers or accepts a stream of characters. The individual characters are not addressable. For example printers, keyboards etc.
- **Network device:** For transmitting data packets

Functions of Device Management

The main functions of device management in the operating system

- An operating system manages communication with the devices through their respective drivers. The operating system component provides a uniform interface to access devices of varied physical attributes. For device management in operating system:
- Keep tracks of all devices and the program which is responsible to perform this is called I/O controller.
- Monitoring the status of each device such as storage drivers, printers and other peripheral devices.
- Enforcing preset policies and taking a decision which process gets the device when and for how long.
- Allocates and Deallocates the device in an efficient way. De-allocating them at two levels: at the process level when I/O command has been executed and the device is temporarily released, and at the job level, when the job is finished and the device is permanently released.
- Optimizes the performance of individual devices.

Types of Devices

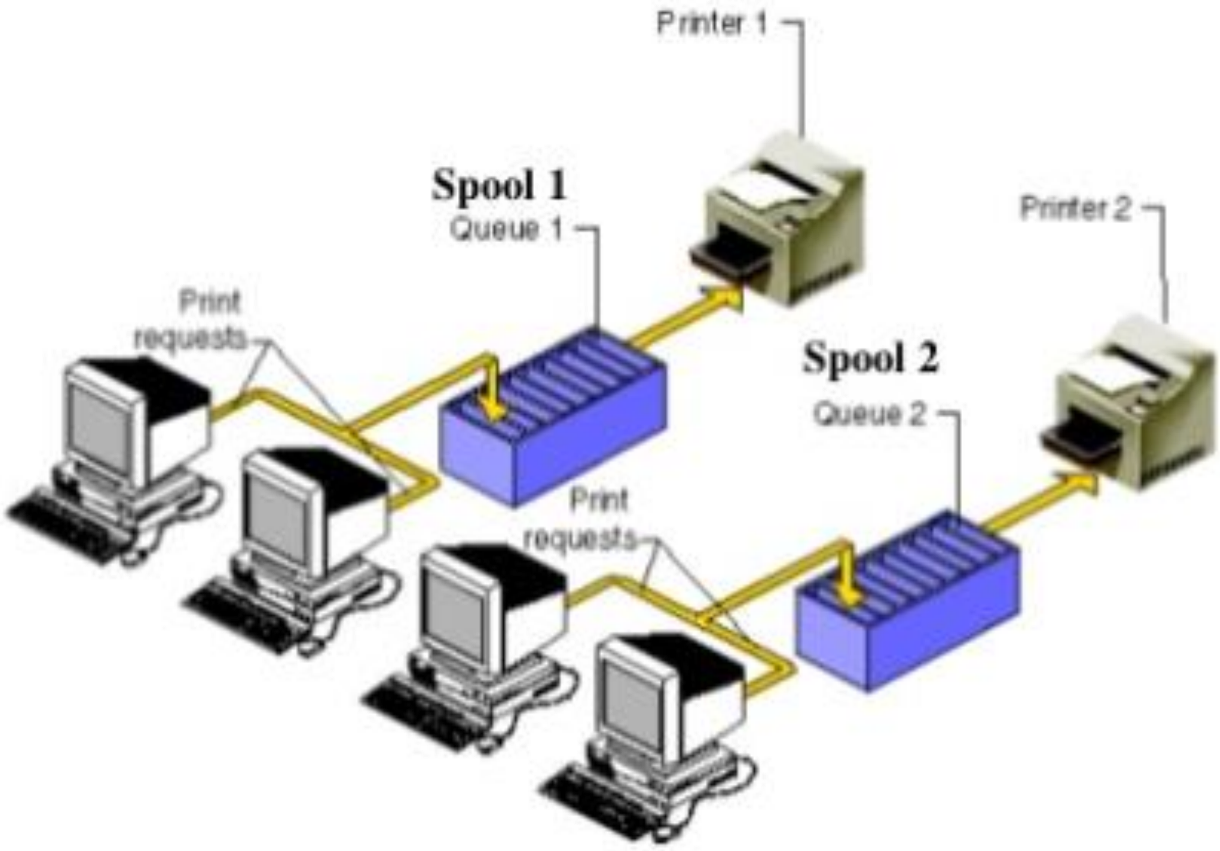
Types of devices: The OS peripheral devices can be categorized into : Dedicated, Shared, and Virtual. They are managed by the Device Manager.

- **Dedicated devices:-** Such type of devices in the **device management in operating system** are dedicated or assigned to only one job at a time until that job releases them. Devices like printers, tape drivers, plotters etc. demand such allocation scheme since it would be awkward if several users share them at the same point of time. The disadvantages of such kind of devices is the inefficiency resulting from the allocation of the device to a single user for the entire duration of job execution even though the device is not put to use 100% of the time.

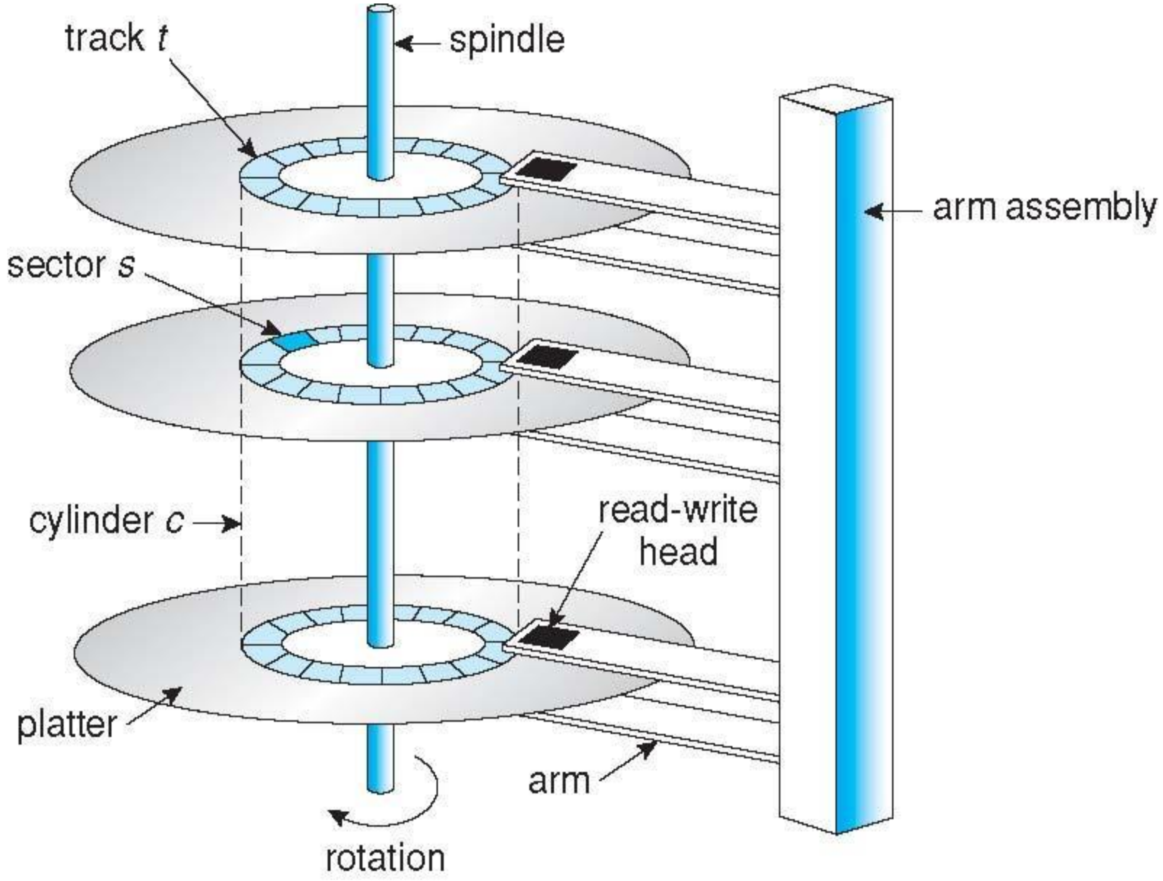
Types of Devices

- **Shared devices:-** These devices can be allocated to several processes. Hard Disk can be shared among several processes at the same time by interleaving their requests. The interleaving is carefully controlled by the Device Manager and all issues must be resolved on the basis of predetermined policies.
- **Virtual Devices:-** These devices are the combination of the first two types and they are dedicated devices which are transformed into shared devices. For example, a printer converted into a shareable device via spooling program which re-routes all the print requests to a disk. A print job is not sent straight to the printer, instead, it goes to the disk(spool) until it is fully prepared with all the necessary sequences and formatting, then it goes to the printers. This technique can transform one printer into several virtual printers which leads to better performance and use.

Types of Devices



Overview of Mass-Storage Structure: Magnetic Disk



Overview of Mass Storage Structure

- **Magnetic disks** provide bulk of secondary storage of modern computers
  - Drives rotate at 60 to 200 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
- Disks can be removable
- Drive attached to computer via **I/O bus**
  - Busses vary, including **EIDE**, **ATA**, **SATA**, **USB**, **Fibre Channel**, **SCSI**
  - **Host controller** in computer uses bus to talk to **disk controller** built into drive or storage array

Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of **logical blocks**, where the logical block is the smallest unit of transfer

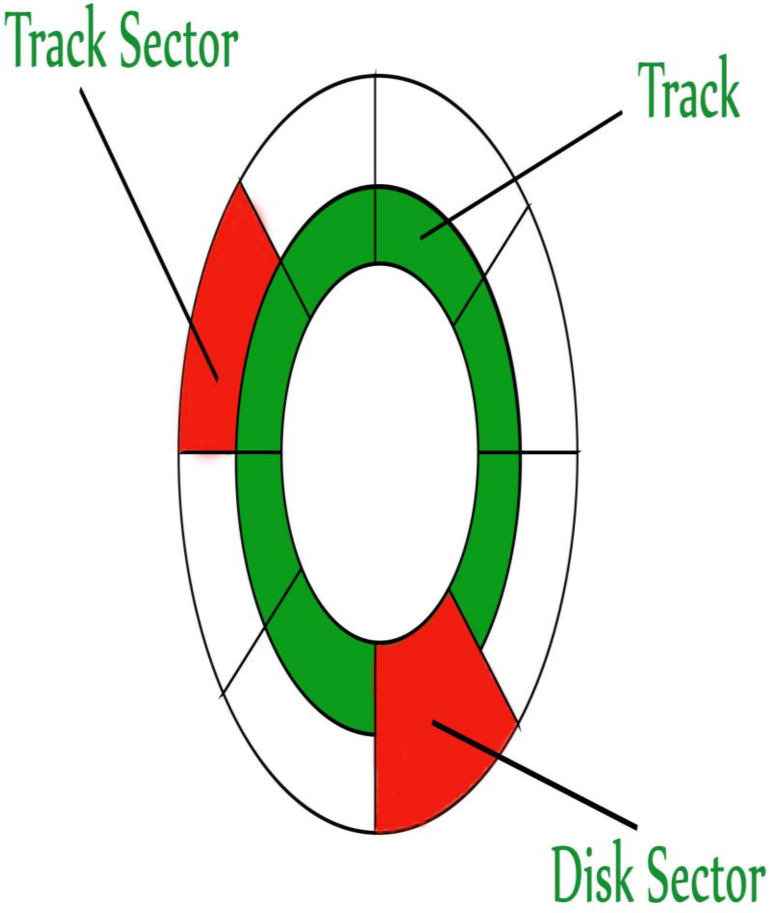
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially
  - Sector 0 is the first sector of the first track on the outermost cylinder
  - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth
- Major components
  - Seek time
  - Rotational latency
  - Transfer Time
  - Disk Access time
  - Disk Response time

- 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



- **Seek Time:** Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or write. So the disk scheduling algorithm that gives minimum average seek time is better.
- **Rotational Latency:** Rotational Latency is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads. So the disk scheduling algorithm that gives minimum rotational latency is better.
- **Transfer Time:** Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred.
- **Disk Access Time:** Disk Access Time is:

Disk Access Time = Seek Time + Rotational Latency + Transfer Time

- **Disk Response Time:** Response Time is the average of time spent by a request waiting to perform its I/O operation.

Numerical Problem

Consider a hard disk with:

4 surfaces

64 tracks/surface

128 sectors/track

256 bytes/sector

Calculate the following

1.What is the capacity of the hard disk?

2.The disk is rotating at 3600 RPM, what is the data transfer rate?

3.The disk is rotating at 3600 RPM, what is the average access time?

SOLUTION

1.What is the capacity of the hard disk?

Disk capacity = surfaces * tracks/surface * sectors/track * bytes/sector

Disk capacity = 4 * 64 * 128 * 256

Disk capacity = 8 MB

2.The disk is rotating at 3600 RPM, what is the data transfer rate?

60 sec -> 3600 rotations

1 sec -> 60 rotations

Data transfer rate = number of rotations per second * track capacity * number of surfaces (since 1 R-W head is used for each surface)

Data transfer rate = 60 * 128 * 256 * 4

Data transfer rate = 7.5 MB/sec

3.The disk is rotating at 3600 RPM, what is the average access time?

Since, seek time, controller time and the amount of data to be transferred is not given, we consider all the three terms as 0.

Therefore, Average Access time = Average rotational delay

Rotational latency => 60 sec -> 3600 rotations

1 sec -> 60 rotations

Rotational latency = (1/60) sec = 16.67 msec.

Average Rotational latency = (16.67)/2

= 8.33 msec.

Average Access time = 8.33 msec.

Solid-State Disks(SSD)

- As technologies improve and economics change, old technologies are often used in different ways. One example of this is the increasing use of solid state disks, or SSDs.
- SSDs use memory technology as a small fast hard disk. Specific implementations may use either flash memory or DRAM chips protected by a battery to sustain the information through power cycles.
- Because SSDs have no moving parts they are much faster than traditional hard drives, and certain problems such as the scheduling of disk accesses simply do not apply.
- However SSDs also have their weaknesses: They are more expensive than hard drives, generally not as large, and may have shorter life spans.
- SSDs are also used in laptops to make them smaller, faster, and lighter.
- Because SSDs are so much faster than traditional hard disks, the throughput of the bus can become a limiting factor, causing some SSDs to be connected directly to the system PCI bus for example.

Solid-State Disks(SSD)

Magnetic Tapes

- Magnetic tapes were once used for common secondary storage before the days of hard disk drives, but today are used primarily for backups.
- Accessing a particular spot on a magnetic tape can be slow, but once reading or writing commences, access speeds are comparable to disk drives.
- Capacities of tape drives can range from 20 to 200 GB, and compression can double that capacity**.**

Disk Scheduling Algorithms

- Several algorithms exist to schedule the servicing of disk I/O requests
- We illustrate them with a request queue (0-199)

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

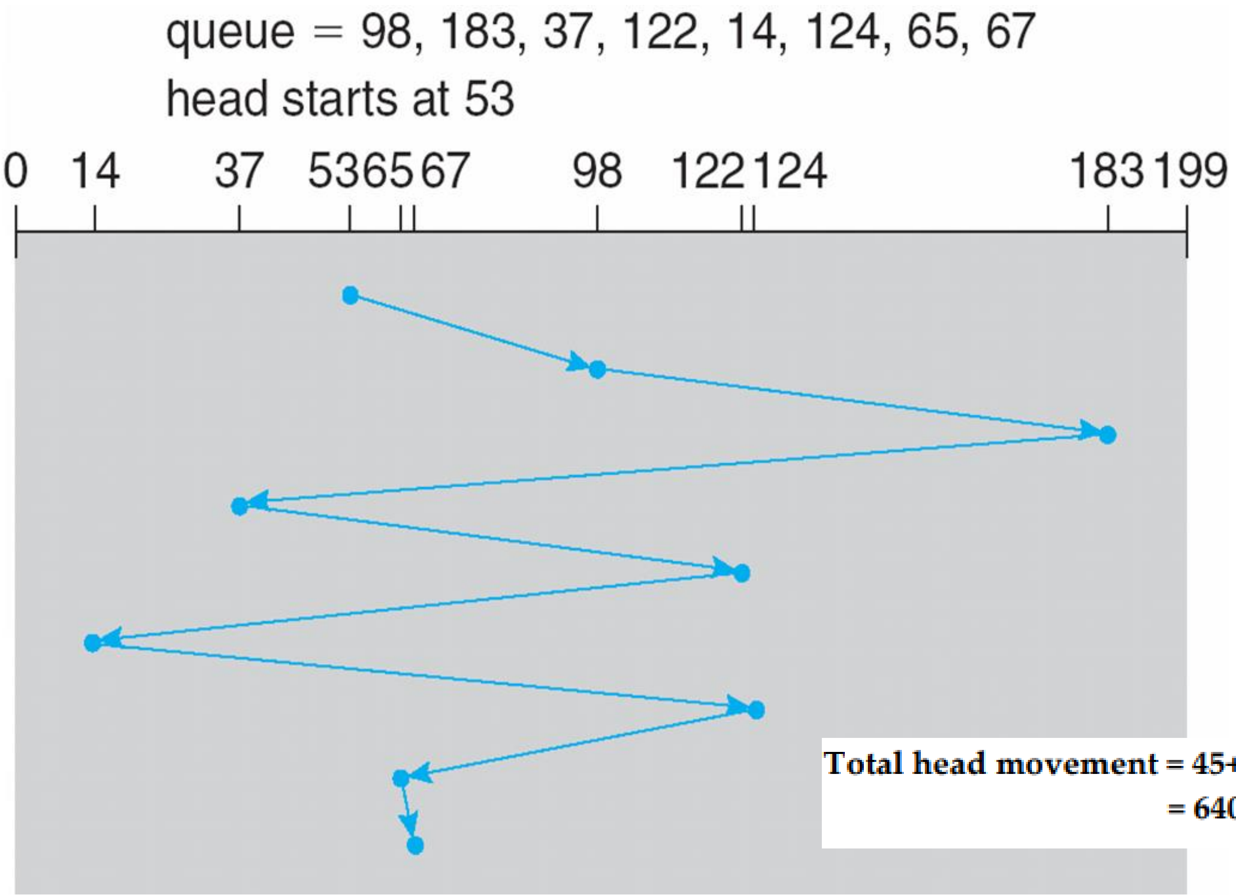
Head pointer 53

FCFS

- FCFS stands for first come first serve.
- It will process the request in the same order in which they were made.
- The request made first will be executed first.
- It is similar to First come First Serve Scheduling Algorithm.

FCFS

Illustration shows total head movement of 640 cylinders



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

head starts at 53

Total head movement = 45+85+146+85+108+110+59+2
= 640

FCFS

Advantages :

- Simple , fair to all request.
- No starvation.

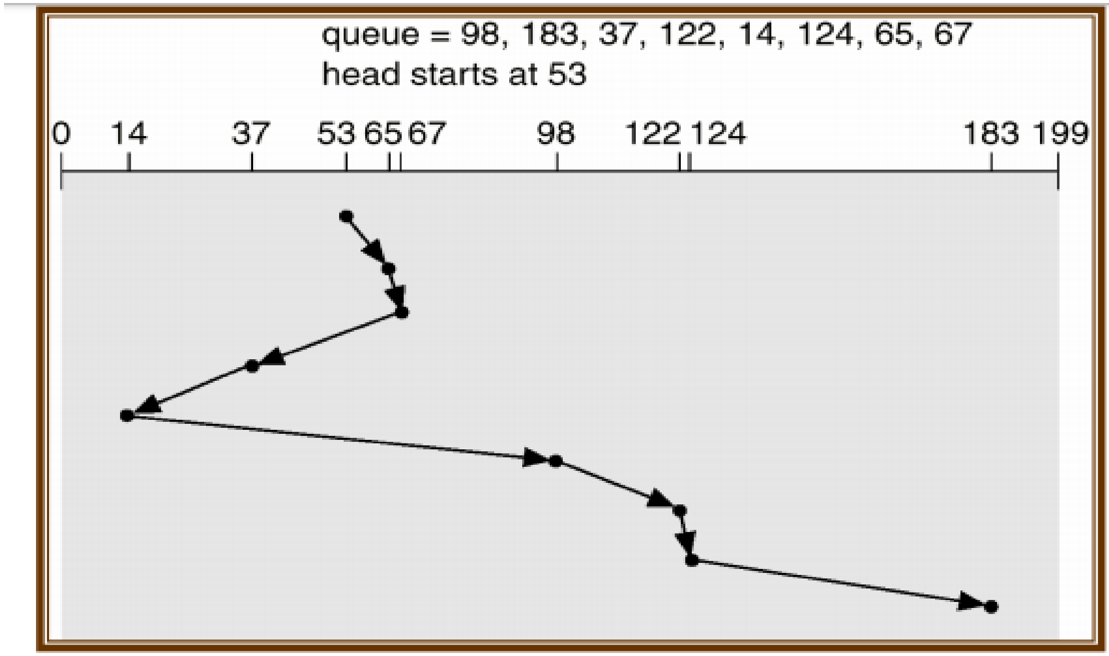
Disadvantages :

- Not efficient because the average seek time is high.

Shortest Seek Time First (SSTF)

- 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

SSTF (Cont.)



Total head movement = 12+2+30+23+84+24+2+59

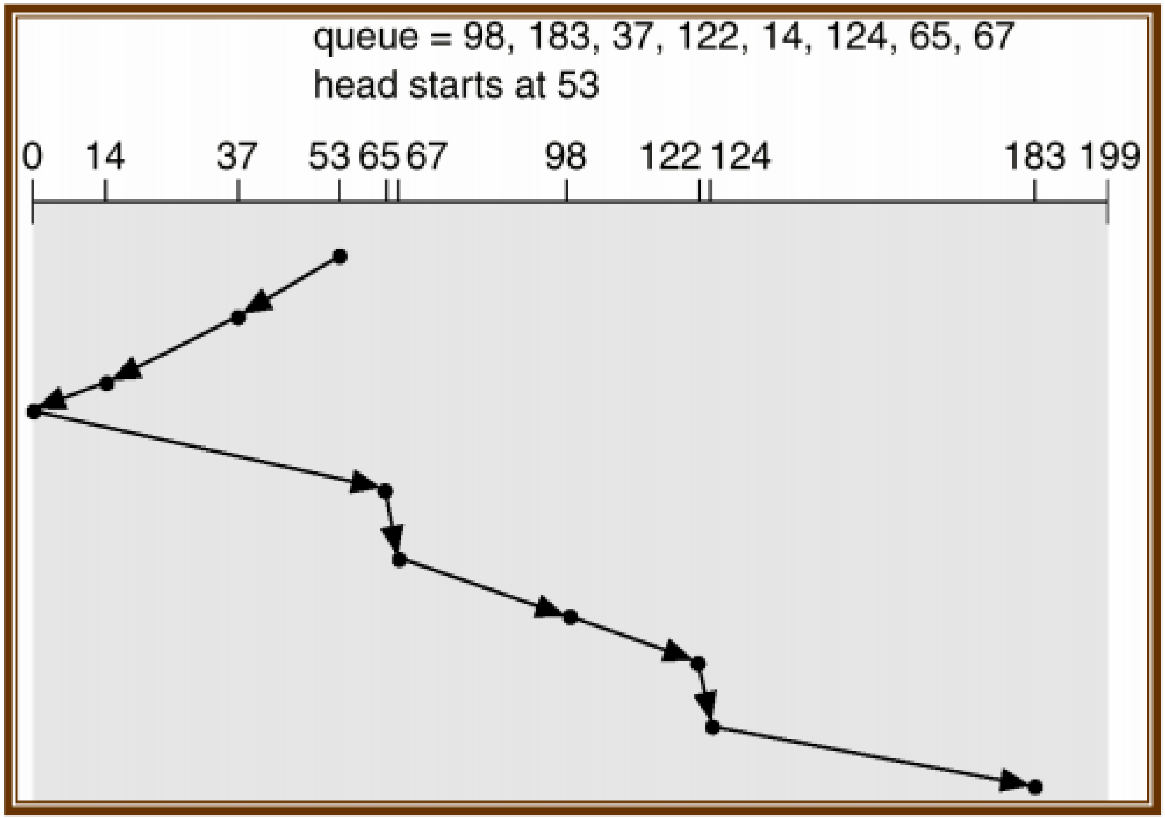
= 236

- Advantages :
  - More efficient than FCFS.
- Disadvantages :
  - Starvation is possible for request involving longer seek-time.

SCAN Scheduling

- 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**
- The head starts at the one end of the disk and moves toward on the other end. It serves all the requests coming in the way.
- After reaching another end the direction of head movement is reverse.

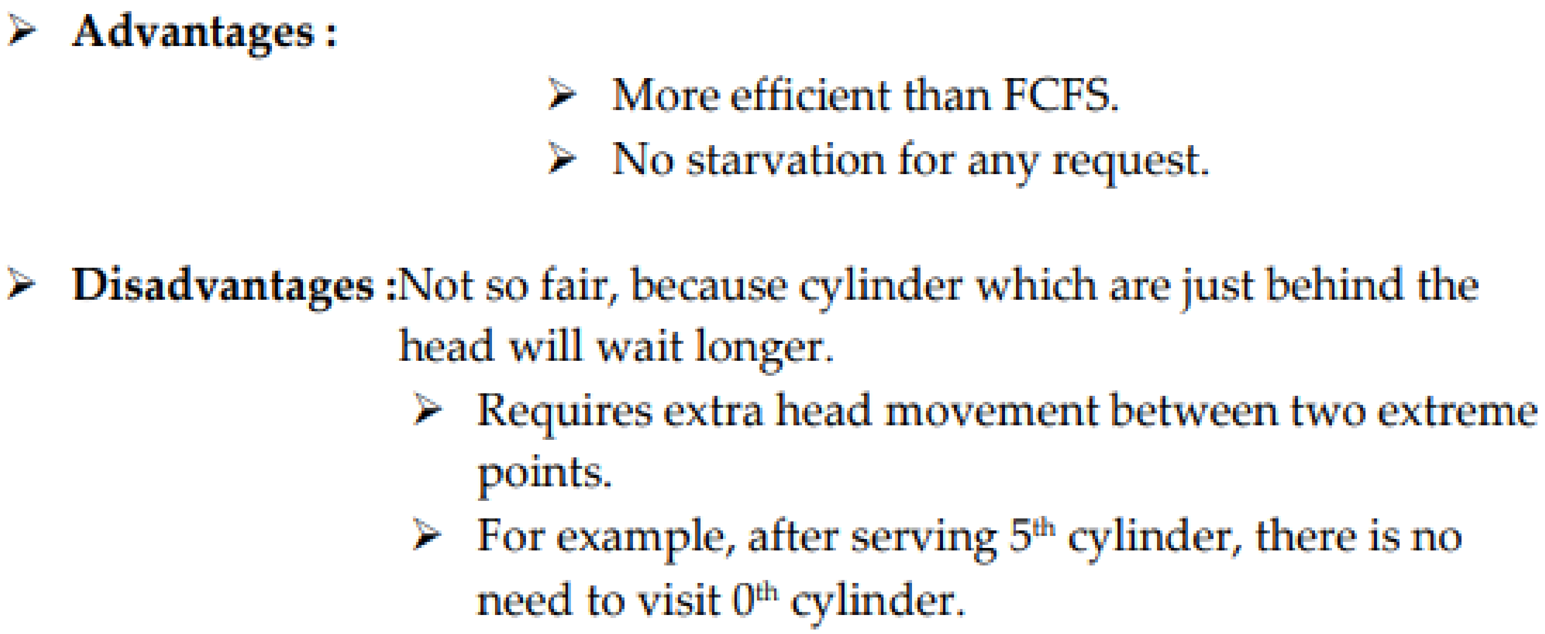
SCAN (Cont.)



Total head movement = 16+23+14+65+2+31+24+2+59

= 236

SCAN Scheduling



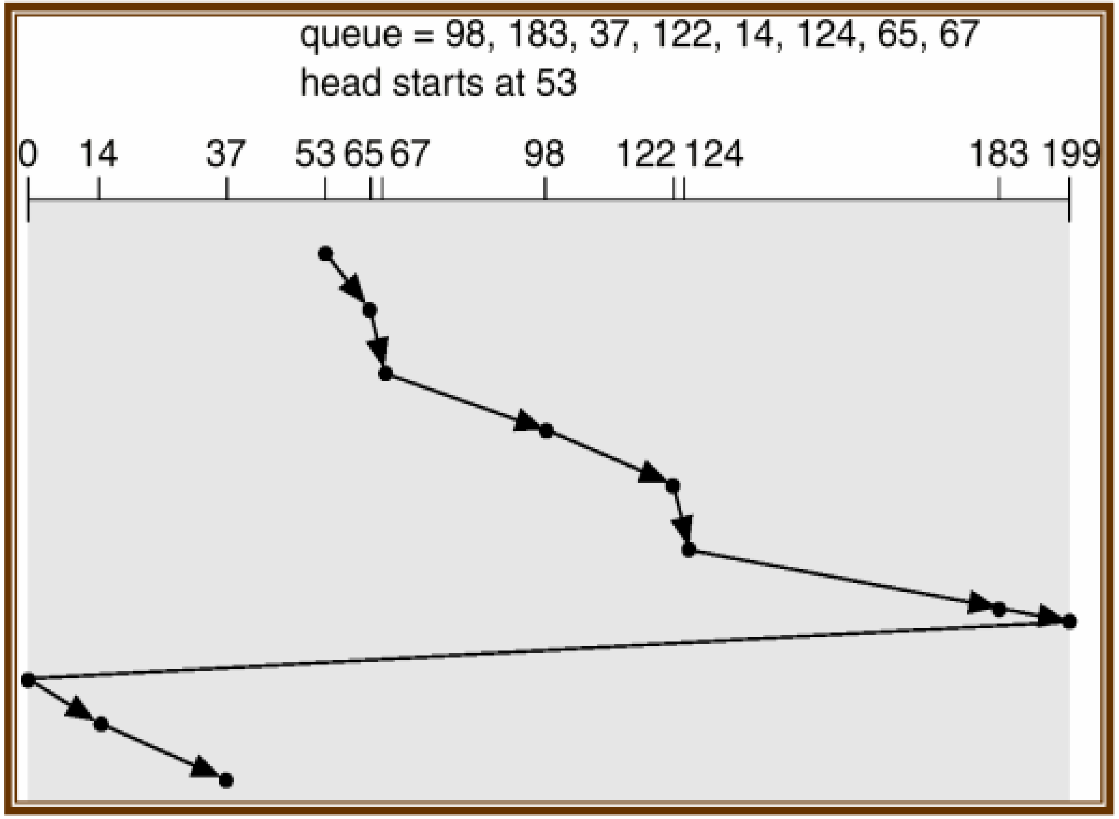
- **Advantages :**
  - More efficient than FCFS.
  - No starvation for any request.

- **Disadvantages :**Not so fair, because cylinder which are just behind the head will wait longer.
  - Requires extra head movement between two extreme points.
  - For example, after serving 5th cylinder, there is no need to visit 0th cylinder.

C-SCAN [Circular SCAN] SCHEDULING

- The circular SCAN algorithm improves upon SCAN by treating all the request in a circular queue fashion.- once the head reaches the end of the disk, it returns to the other end without processing any request, and then starts again from the beginning of the disk.
- 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.
- Illustration shows total head movement of 382 cylinders, plus return time.
- Provides a more uniform wait time than SCAN.

C-SCAN (Cont.)



Total head movement = 12+2+31+24+2+59+16+199+14+23
= 382

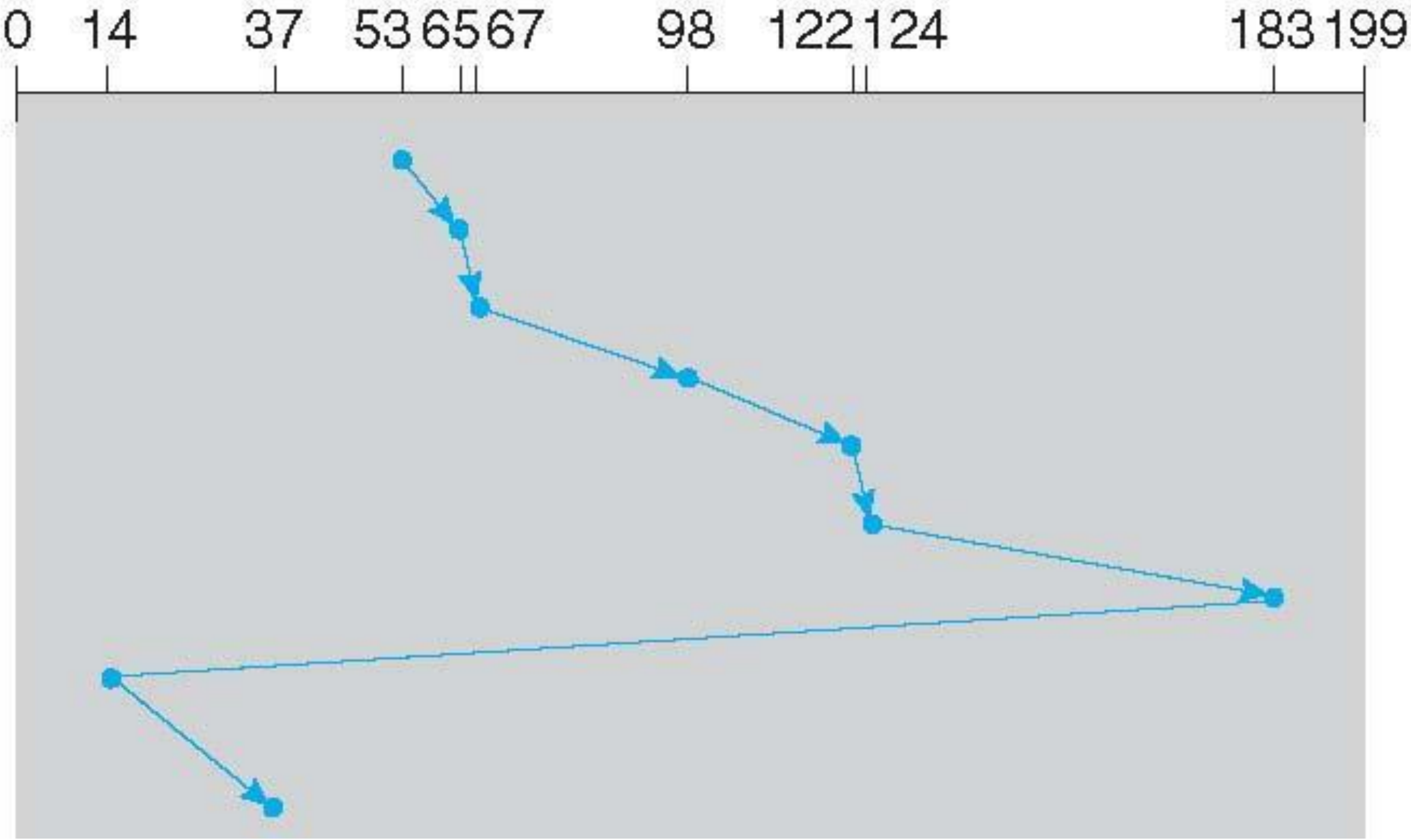
C-LOOK SCAN

- 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
- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file-allocation method
- 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

Example

Consider a disk with 200 tracks and the queue has random requests from different processes in the order:

55, 58, 39, 18, 90, 160, 150, 38, 184

Initially arm is at 100. Find the Average Seek length using FIFO, SSTF, SCAN and C-SCAN algorithm.

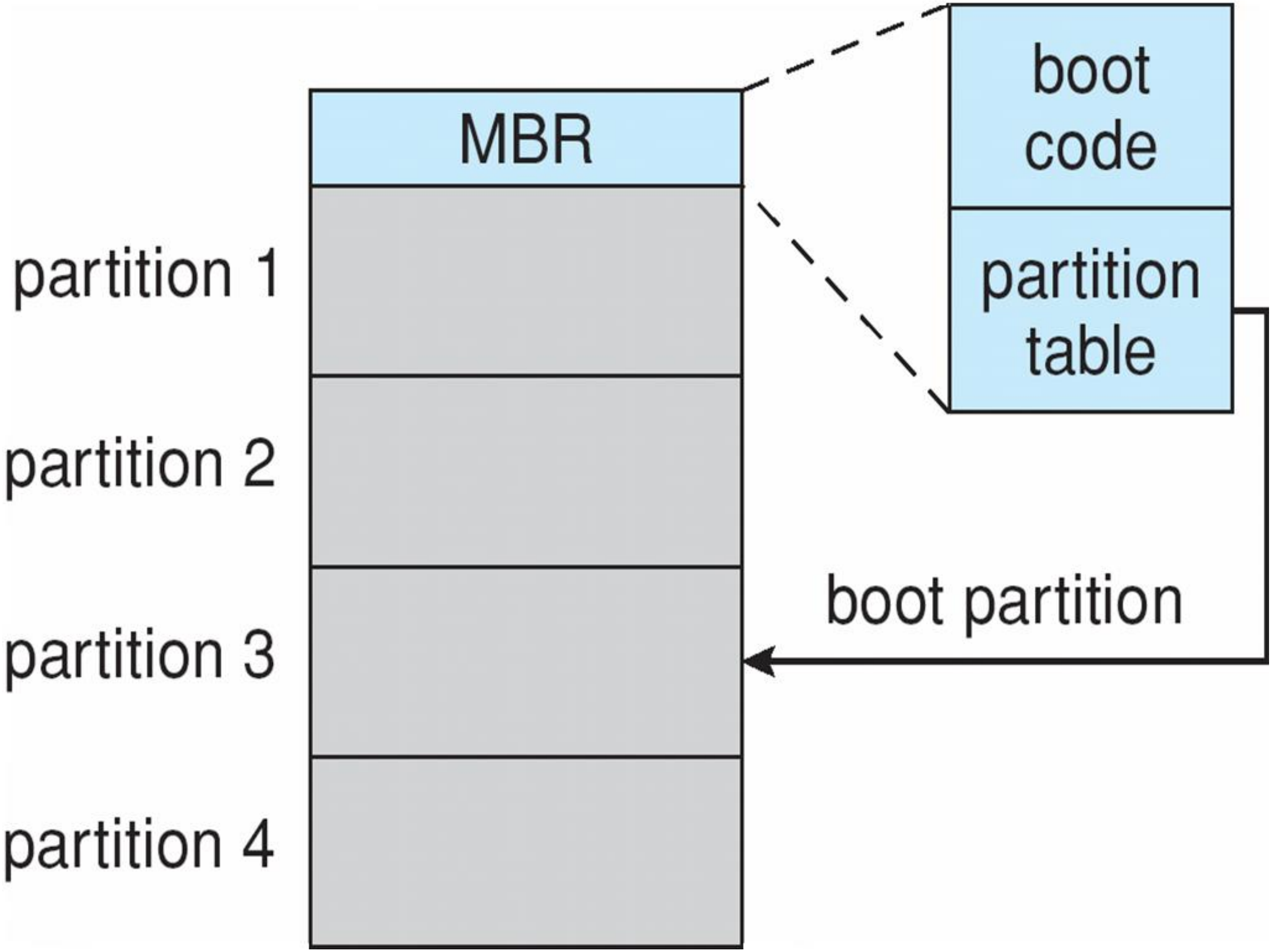
Example

Using FCFS		Using SSTF		Using SCAN		Using C - SCAN	
Next request to be serviced	No. of cylinders travelled	Next request to be serviced	No. of cylinders travelled	Next request to be serviced	No. of cylinders travelled	Next request to be serviced	No. of cylinders travelled
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average Seek Length	498 / 9 = 55.3	Average Seek Length	27.5	Average Seek Length	27.8	Average Seek Length	35.6

Disk Management

- **Low-level formatting**, or **physical formatting** — Dividing a disk into sectors that the disk controller can read and write
- 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
  - **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
- Boot block initializes system
  - The bootstrap is stored in ROM
  - **Bootstrap loader** program

Booting from a Disk in



RAID (Redundant Array of Independent Disks)

- RAID or Redundant Array of Independent Disks, is a technology to connect multiple secondary storage devices and use them as a single storage media.

- RAID consists of an array of disks in which multiple disks are connected together to achieve different goals. RAID levels define the use of disk arrays.

- A variety of disk-organization techniques, collectively called redundant arrays of independent disks (RAID), are commonly used to address the performance and reliability issues.
- In the past, RAIDs composed of small, cheap disks were viewed as a cost-effective alternative to large, expensive disks. Today, RAIDs are used for their higher reliability and higher data-transfer rate, rather than for economic reasons.

Improvement of Reliability via Redundancy

- The simplest (but most expensive) approach to introducing redundancy is to duplicate every disk. This technique is called mirroring.
- With mirroring, a logical disk consists of two physical disks, and every write is carried out on both disks.
- The result is called a mirrored volume. If one of the disks in the volume fails, the data can be read from the other. Data will be lost only if the second disk fails before the first failed disk is replaced.

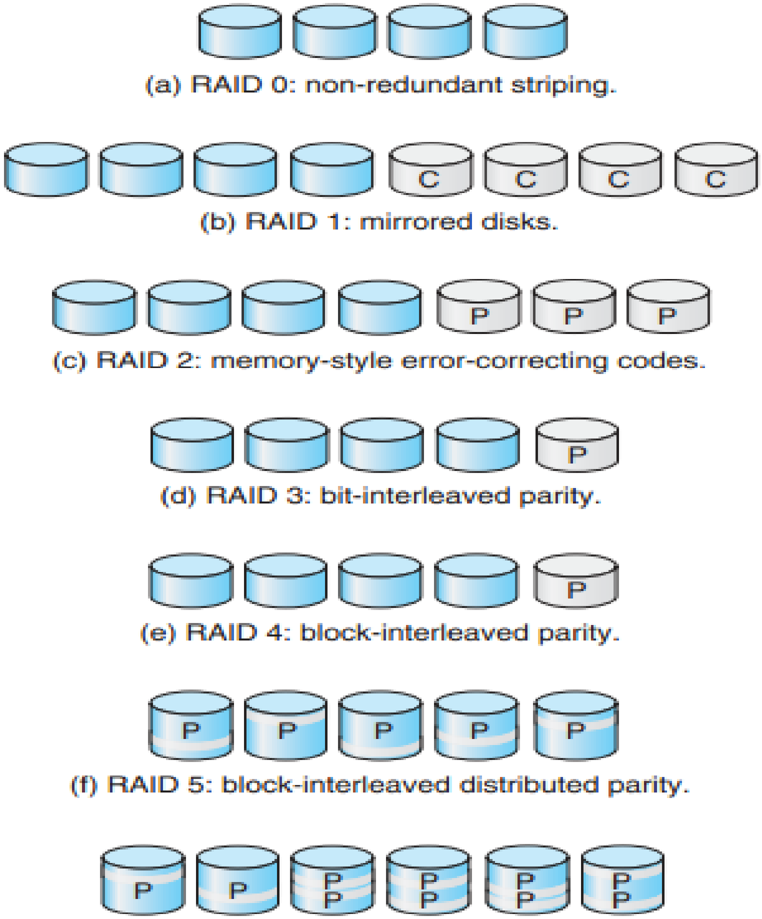
Improvement in Performance via Parallelism

- With multiple disks, we can improve the transfer rate as well (or instead) by striping data across the disks. In its simplest form, data striping consists of splitting the bits of each byte across multiple disks; such striping is called **bit-level striping.**
- In **block-level striping,** for instance, blocks of a file are striped across multiple disks; with n disks, block i of a file goes to disk (i mod n) + 1. Other levels of striping, such as bytes of a sector or sectors of a block, also are possible. Block-level striping is the most common.

Parallelism in a disk system, as achieved through striping, has two main goals:

- 1. Increase the throughput of multiple small accesses (that is, page accesses) by load balancing.
- 2. Reduce the response time of large accesses

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.

Figure 10.11 RAID levels.

RAID LEVELS

RAID level 0: RAID level 0 refers to disk arrays with striping at the level of blocks but without any redundancy (such as mirroring or parity bits).

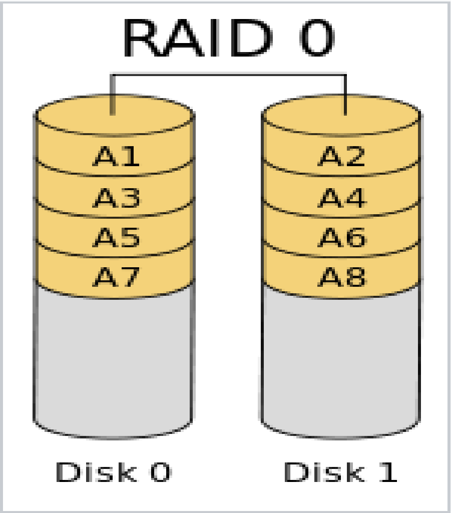


Diagram of a RAID 0 setup

RAID LEVEL 0

Advantages

- RAID 0 offers great performance, both in read and write operations. There is no overhead caused by parity controls.
- All storage capacity is used, there is no overhead.
- The technology is easy to implement.

Disadvantages

- RAID 0 is not fault-tolerant. If one drive fails, all data in the RAID 0 array are lost. It should not be used for mission-critical systems.

RAID LEVEL 1

RAID level 1 refers to disk mirroring.

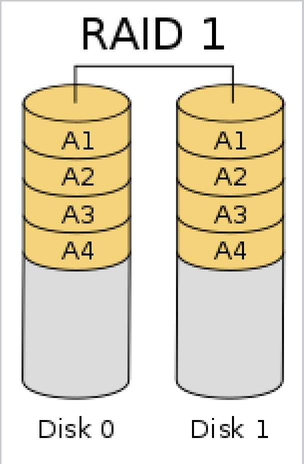


Diagram of a RAID 1 setup

RAID LEVEL 1

Advantages

- RAID 1 offers excellent read speed and a write-speed that is comparable to that of a single drive.
- In case a drive fails, data do not have to be rebuild, they just have to be copied to the replacement drive.
- RAID 1 is a very simple technology.

Disadvantages

- The main disadvantage is that the effective storage capacity is only half of the total drive capacity because all data get written twice.
- Software RAID 1 solutions do not always allow a hot swap of a failed drive. That means the failed drive can only be replaced after powering down the computer it is attached to. For servers that are used simultaneously by many people, this may not be acceptable. Such systems typically use hardware controllers that do support hot swapping.

RAID LEVEL 2

RAID level 2 is also known as memory-style error-correcting-code (ECC) organization. Memory systems have long detected certain errors by using parity bits. Each byte in a memory system may have a parity bit associated with it that records whether the number of bits in the byte set to 1 is even (parity = 0) or odd (parity = 1). If one of the bits in the byte is damaged (either a 1 becomes a 0, or a 0 becomes a 1), the parity of the byte changes and thus does not match the stored parity. Similarly, if the stored parity bit is damaged, it does not match the computed parity. Thus, all single-bit errors are detected by the memory system.

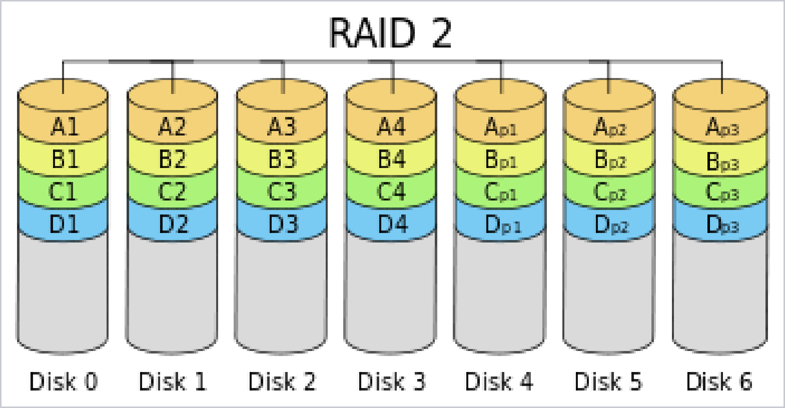


Diagram of a RAID 2 setup

RAID LEVEL 3

RAID level 3, or bit-interleaved parity organization, improves on level 2 by taking into account the fact that, unlike memory systems, disk controllers can detect whether a sector has been read correctly, so a single parity bit can be used for error correction as well as for detection. The idea is as follows: If one of the sectors is damaged, we know exactly which sector it is, and we can figure out whether any bit in the sector is a 1 or a 0 by computing the parity of the corresponding bits from sectors in the other disks. If the parity of the remaining bits is equal to the stored parity, the missing bit is 0; otherwise, it is 1. RAID level 3 is as good as level 2 but is less expensive in the number of extra disks required (it has only a one-disk overhead).

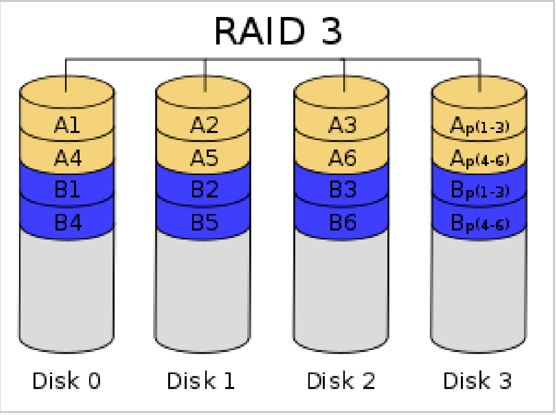


Diagram of a RAID 3 setup of six-byte blocks and two parity bytes, shown are two blocks of data in different colors.

RAID LEVEL 4

- RAID 4 consists of block-level striping with a dedicated parity disk. As a result of its layout, RAID 4 provides good performance of random reads, while the performance of random writes is low due to the need to write all parity data to a single disk.
- In diagram 1, a read request for block A1 would be serviced by disk 0. A simultaneous read request for block B1 would have to wait, but a read request for B2 could be serviced concurrently by disk 1.

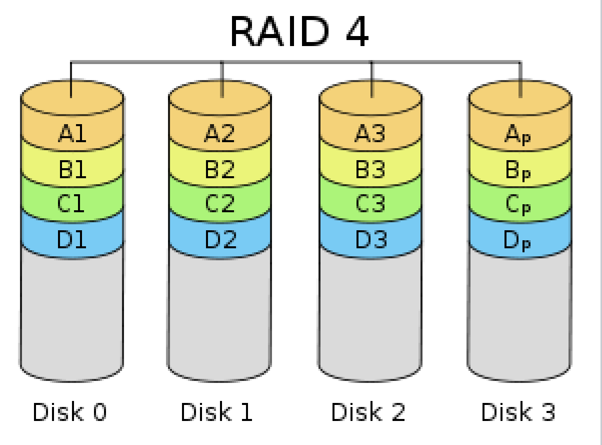


Diagram 1: A RAID 4 setup with dedicated parity disk with each color representing the group of blocks in the respective parity block (a stripe)

RAID LEVEL 5

- RAID 5 writes whole data blocks onto different disks, but the parity bits generated for data block stripe are distributed among all the data disks rather than storing them on a different dedicated disk.
- RAID 5 consists of block-level striping with distributed parity. Unlike in RAID 4, parity information is distributed among the drives. It requires that all drives but one be present to operate. Upon failure of a single drive, subsequent reads can be calculated from the distributed parity such that no data is lost. RAID 5 requires at least three disks.

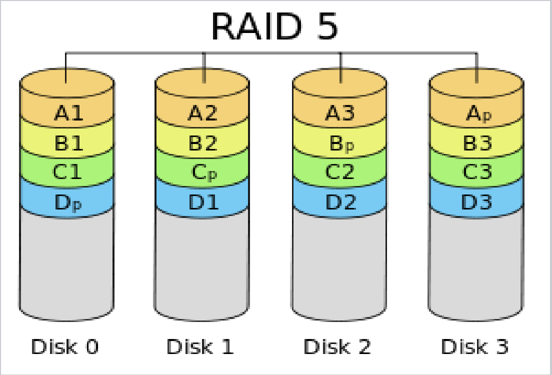


Diagram of a RAID 5 setup with distributed parity with each color representing the group of blocks in the respective parity block (a stripe). This diagram shows left asymmetric algorithm

RAID LEVEL 5

Advantages

- Read data transactions are very fast while write data transactions are somewhat slower (due to the parity that has to be calculated).
- If a drive fails, you still have access to all data, even while the failed drive is being replaced and the storage controller rebuilds the data on the new drive.

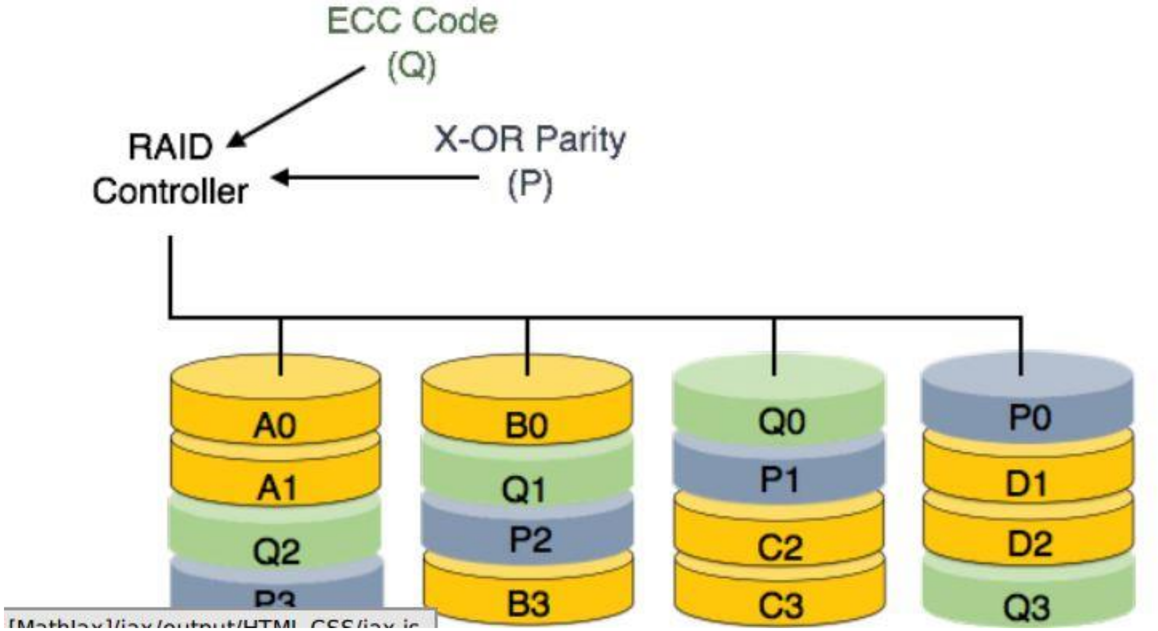
Disadvantages

- Drive failures have an effect on throughput, although this is still acceptable.
- This is complex technology. If one of the disks in an array using 4TB disks fails and is replaced, restoring the data (the rebuild time) may take a day or longer, depending on the load on the array and the speed of the controller. If another disk goes bad during that time, data are lost forever.
- Ideal use

RAID 5 is a good all-round system that combines efficient storage with excellent security and decent performance. It is ideal for file and application servers that have a limited number of data drives.

RAID 6

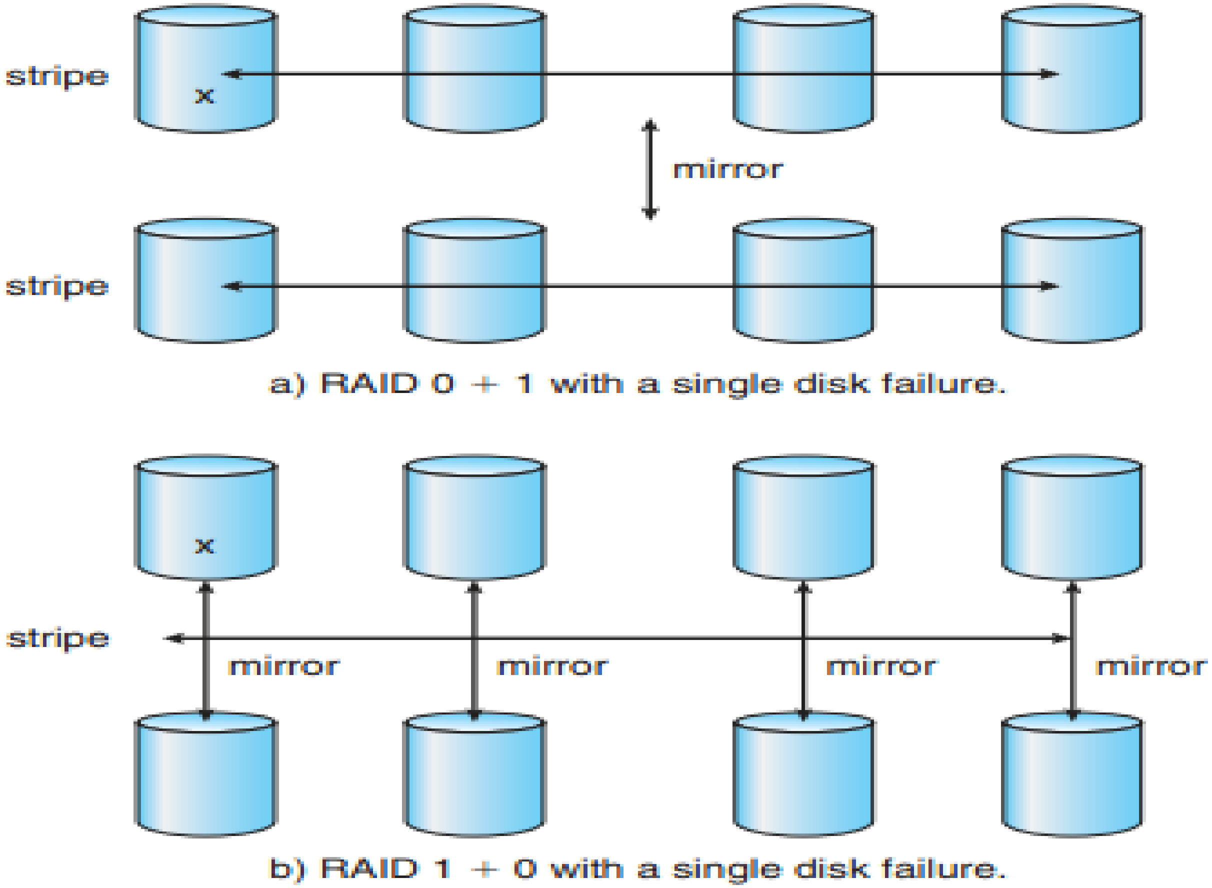
RAID 6 is an extension of level 5. In this level, two independent parities are generated and stored in distributed fashion among multiple disks. Two parities provide additional fault tolerance. This level requires at least four disk drives to implement RAID.



RAID LEVELS

RAID 10

It is also known as RAID 1+0, is a RAID configuration that combines disk mirroring and disk striping to protect data. It requires a minimum of four disks and stripes data across mirrored pairs. As long as one disk in each mirrored pair is functional, data can be retrieved.



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

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

Figure 10.12 RAID 0 + 1 and 1 + 0.

Thank You

For any Query, you can contact

Er. Inderjeet Singh(e8822)
Ph. No: 86-99-100-160
Email id: inderjeet.e8822@cumail.in