UNIT IV DEVICE MANAGEMENT

Types of Devices-Sequential Access Storage Media-Direct Access Storage Devices- Magnetic Disk Drive Access Times- Components of the I/O Subsystem- Communication among Devices-Management of I/O Requests.

Device management involves four basic functions:

- Monitoring the status of each device, such as storage drives, printers, and other peripheral devices
- Enforcing preset policies to determine which process will get a device and for how long
- Allocating the devices
- Deallocating them at two levels—at the process (or task) level when an I/O command has been executed and the device is temporarily released, and then at the job level when the job is finished and the device is permanently released

TYPES OF DEVICES

The system's peripheral devices generally fall into one of three categories:

1. Dedicated
2. Shared
3. Virtual.

DEDICATED DEVICES

- **Dedicated devices** are assigned to only one job at a time.
- They serve that job for the entire time it is active or until it releases them.
- Some devices, such as tape drives, printers, and plotters, demand this kind of allocation scheme

Disadvantage

- Device must be allocated to a single user for the duration of a job's execution, which can be quite inefficient, especially when the device isn't used 100 percent of the time.

SHARED DEVICES

- **Shared devices** can be assigned to several processes.
- A disk, or any other direct access **storage** device (often shortened to DASD), can be shared by several processes at the same time by interleaving their requests.

- Conflicts occur by interleaving must be resolved based on predetermined policies to decide which request will be handled first.

VIRTUAL DEVICES

- **Virtual devices** are a combination of the first two. They are dedicated devices that have been transformed into shared devices.
- For example, printers (which are dedicated devices) are converted into sharable devices through a spooling program that reroutes all print requests to a disk.
- Only when all of a job's output is complete, and the printer is ready to print out the entire document, is the output sent to the printer for printing.

CLASSIFICATION OF STORAGE MEDIA

Storage media are divided into two groups.

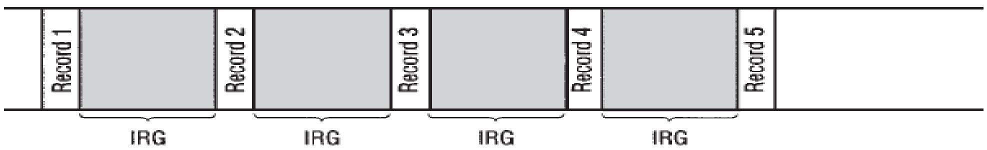
1. **Sequential access media -** which store records sequentially, one after the other.
2. D**irect access storage devices (DASD**) - which can store either sequential or direct access files.

Sequential Access Storage Media

- Magnetic tape was developed for routine secondary storage in early computer systems.
- Records are stored serially, one after the other.
- To access a single record, the tape must be mounted and fast-forwarded from its beginning until the desired position is located. This can be a time-consuming process.
- The number of characters that can be recorded per inch is determined by the density of the tape, such as 1600 bytes per inch (bpi).
- However number of characters per inch would depend on how does the record is stored in tape.
- If the records are **stored individually**, each record would need to be separated by a space to indicate its starting and ending places.
- If the records are **stored in blocks**, then the entire block is preceded by a space and followed by a space, but the individual records are stored sequentially within the block.

Magnetic tape moves under the read/write head only when there's a need to access a record; at all other times it is standing still. The tape needs time and space to stop, so a gap is inserted between each record.

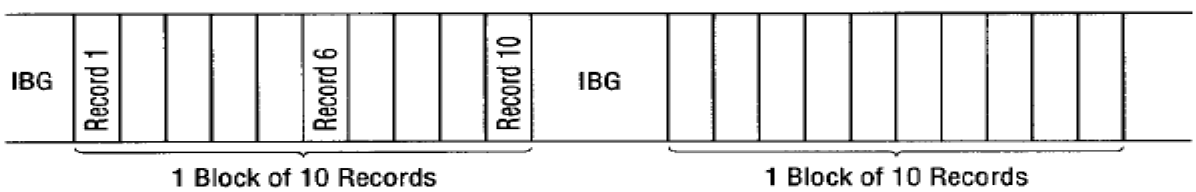
This inter-record gap (IRG) is about 1⁄2 inch long regardless of the sizes of the records it separates. Therefore, if 10 records are stored individually, there will be nine 1⁄2-inch IRGs between each record.



An alternative is to group the records into blocks before recording them on tape. This is called **blocking** and it's performed when the file is created.

The number of records in a block is usually determined by the application program, and it's often set to take advantage of the **transfer rate,** which is the density of the tape (measured in bpi), multiplied by the tape drive speed, called transport speed, which is measured in inches per second (ips):

transfer rate (ips) = density * transport speed



Advantages:

- Fewer I/O operations are needed because a single READ command can move an entire block, the physical record that includes several logical records, into main memory.
- Less tape is wasted because the size of the physical record exceeds the size of the gap.

Disadvantages

- Overhead and software routines are needed for blocking, deblocking, and recordkeeping.
- Buffer space may be wasted if you need only one logical record but must read an entire block to get it.

DIRECT ACCESS STORAGE DEVICES

Direct access storage devices (DASDs) are devices that can directly read or write to a specific place. DASDs can be grouped into three categories.

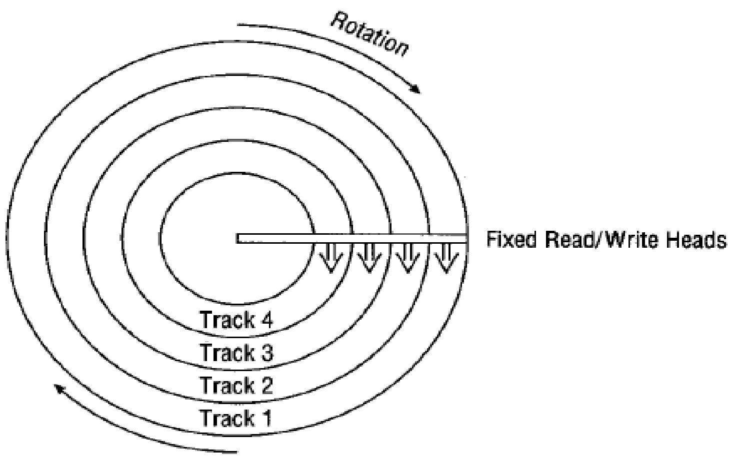
1. Magnetic disks
2. Optical discs
3. Flash memory.

DASD access times isn't as wide as with magnetic tape and the location of the specific record still has a direct effect on the amount of time required to access it.

MAGNETIC DISK

Fixed-Head Magnetic Disk Storage

- A fixed-head magnetic disk looks like a large CD or DVD covered with magnetic film.
- It has been formatted on both sides, into concentric circles.
- Each circle is a **track**. Data is recorded serially on each track by the fixed read/write head positioned over it.
- A fixed-head disk is also very faster than the movable-head disks.



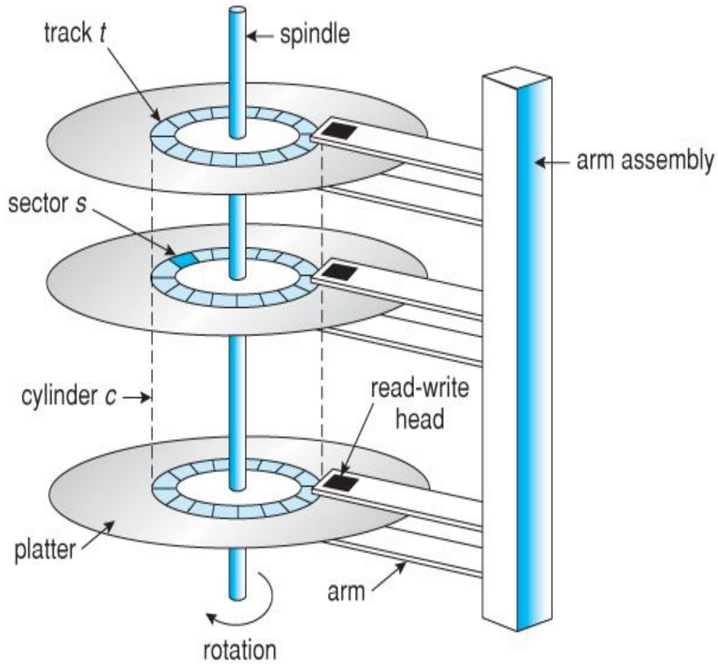
Disadvantages

1. High cost
2. Reduced storage space compared to a movable-head disk (because the tracks must be positioned farther apart to accommodate the width of the read/write heads).
3. These devices have been used when speed is of the utmost importance, such as space flight or aircraft applications.

Movable-Head Magnetic Disk Storage

- Movable-head magnetic disks, such as computer hard drives, have one read/write head that floats over each surface of each disk.
- Disks can be a single platter, or a stack of magnetic platters.

- Several platters are stacked on a common central spindle, separated by enough space to allow the read/write heads to move between each pair of disks.



- Each platter (except those at the top and bottom of the stack) has two surfaces for recording, and each surface is formatted with a specific number of concentric **tracks** where the data is recorded.
- The number of tracks varies from manufacturer to manufacturer, but typically there are a thousand or more on a high-capacity hard disk.
- Each track on each surface is numbered: Track 0 identifies the outermost concentric circle on each surface; the highest-numbered track is in the center.

There are two ways to fill records in disk pack such as

- **Surface-by-Surface**

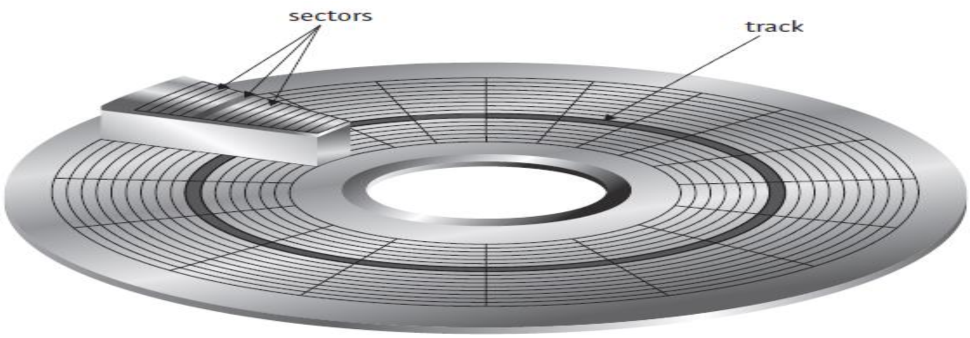
  (write a series of records on surface one when that surface is full, to continue writing on surface two, and then on surface three, and so on)
- **Track-by-Track**

  (Fill up every outside track of every surface before moving the heads inward to the next track position to continue writing)
- It is slower to fill a disk pack surface-by-surface than it is to fill it up track-by-track and this leads us to a valuable concept.

- If Track 0 of all of the surfaces is filled then it forms a virtual **cylinder** of data. There are as many cylinders as there are tracks, and the cylinders are as tall as the disk pack.

To access any given record, the system needs three things:

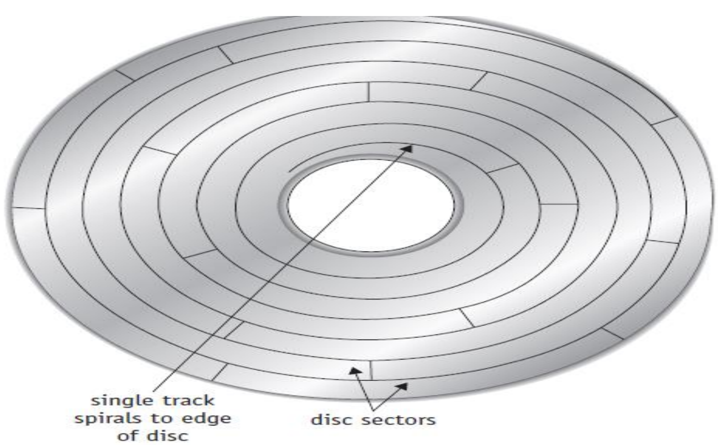
1. Cylinder number, so the arm can move the read/write heads to it;
2. Surface number, so the proper read/write head is activated;
3. Sector number, so the read/write head knows the instant when it should begin reading or writing.



OPTICAL DISC STORAGE

Since a magnetic disk, which consists of concentric tracks of sectors, spins at a constant speed the sectors at the outside of the disk spin faster than the inner sectors; outside sectors are much larger than sectors located near the center of the disk. This format **wastes storage space** but maximizes the speed with which data can be retrieved.

The advent of optical disc storage was made possible by development in laser technology. An optical disc consists of a single spiraling track of same-sized sectors running from the center to the rim of the disc

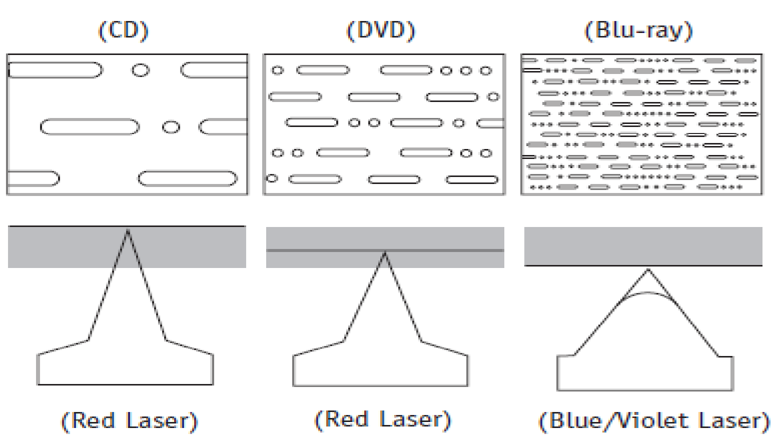


- This single track also has sectors, but all sectors are the same size regardless of their locations on the disc. This design allows many more sectors, and much more data, to fit on an optical disc compared to a magnetic disk of the same size.
- The disc drive adjusts the speed of the disc's spin to compensate for the sector's location on the disc this is called constant linear velocity (CLV). Therefore, the disc spins faster to read sectors located at the center of the disc, and slower to read sectors near the outer edge.

Two of the most important measures of optical disc drive performance are sustained data transfer rate and average access time.

- **The data transfer rate** is measured in **megabytes** per second and refers to the speed at which huge amounts of data can be read from the disc.
- **Access time,** which indicates **the average time required to move the head to a specific place** on the disc, is expressed in **milliseconds (ms)**. The fastest units have the smallest average access time, which is the most important factor when searching for information randomly, such as in a database.
- **Cache size** also has a substantial impact on perceived performance. Hardware cache acts as a buffer by transferring blocks of data from the disc, anticipating that the user may want to reread some recently retrieved information, which can be done quickly if the information remains in the cache.

There are several types of optical-disc systems, depending on the medium and the capacity of the discs: CDs, DVDs, and Blu-ray.



To put data on an optical disc, a high-intensity laser beam burns indentations on the disc that are called pits. These **pits,** which represent 0s, contrast with the unburned flat areas, called **lands,** which represent 1s.

The first sectors are located in the center of the disc and the laser moves outward reading each sector in turn. If a disc has multiple layers, the laser's course is reversed to read the second layer with the arm moving from the outer edge to the inner.

CD and DVD Technology

- In the CD or DVD player, data is read back by focusing a low-powered red laser on it, which shines through the protective layer of the disc onto the CD track (or DVD tracks) where data is recorded.
- Light striking a land is reflected into a photo-detector while light striking a pit is scattered and absorbed. The photo-detector then converts the intensity of the light into a digital signal of 1s and 0s.
- Recordable CD and DVD disc drives require more expensive disc controllers than the read-only disc players because they need to incorporate write mechanisms specific to each medium.
- The software used to create a recordable CD (CD-R) uses a standard format which automatically checks for errors and creates a table of contents, used to keep track of each file's location.
- Similarly, recordable and rewritable CDs (CD-RWs) use a process called phase change technology to write, change, and erase data.
- Although DVDs use the same design and are the same size and shape as CDs, they can store much more data. A dual-layer, single-sided DVD can hold the equivalent of 13 CDs; its red laser, with a shorter wavelength than the CD's red laser, makes smaller pits and allows the spiral to be wound tighter.
- When the advantages of compression technology are added to the high capacity of a DVD, such as MPEG video compression, a single-sided, double-layer DVD can hold 8.6GB, more than enough space to hold a two-hour movie with enhanced audio.

Blu-ray Disc Technology

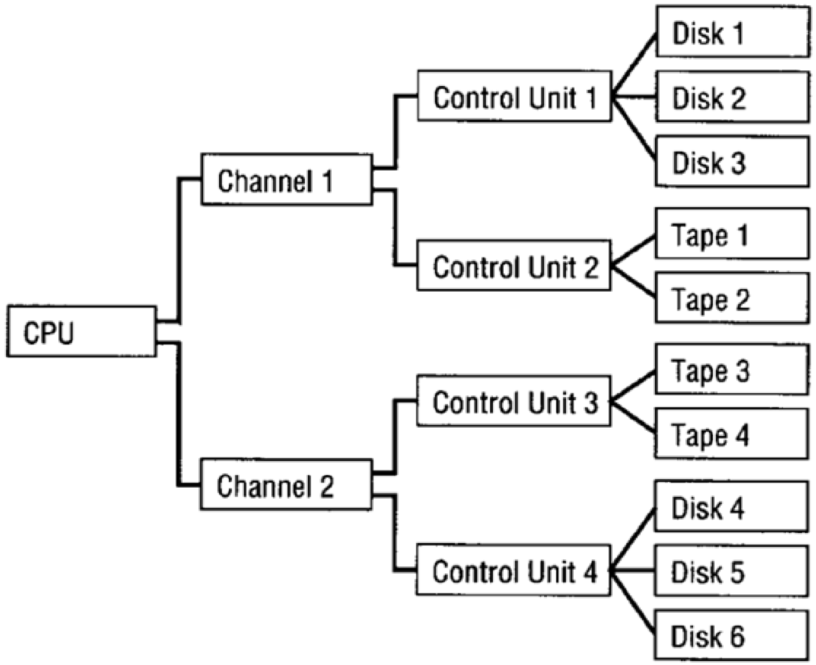
- A Blu-ray disc is the same physical size as a DVD or CD but the laser technology used to read and write data is quite different.

- The pits (each representing a 1) on a Blu-ray disc are much smaller and the tracks are wound much tighter than they are on a DVD or CD.
- Although Blu-ray products can be made backward compatible so they can accommodate the older CDs and DVDs, the Blu-ray's blue-violet laser (405nm) has a shorter wavelength than the CD/DVD's red laser (650nm). This allows data to be packed more tightly and stored in less space.
- The blue-violet laser can write on a much thinner layer on the disc, allowing multiple layers to be written on top of each other and vastly increasing the amount of data that can be stored on a single disc.
- The disc's format was created to further the commercial prospects for high-definition video, and to store large amounts of data, particularly for games and interactive applications via the Java programming language. Blu-ray players execute Java programs for menus and user interaction.
- Each Blu-ray disc can hold much more data (50GB for a two-layer disc) than can a similar DVD (8.5GB for a two-layer disc).
- Reading speed is also much faster with the fastest Blu-ray players featuring 432 Mbps (comparable DVD players reach 168.75 Mbps).
- Like CDs and DVDs, Blu-ray discs are available in several for-mats: read-only (BD-ROM), recordable (BD-R), and rewritable (BD-RE).

Flash Memory Storage

- **Flash memory** is a type of electrically erasable programmable read-only memory (EEP-ROM).
- It's a nonvolatile removable medium that emulates random access memory, but, unlike RAM, stores data securely even when it's removed from its power source.
- Flash memory was primarily used to store startup (boot up) information for computers, but is now used to store data for cell phones, mobile devices, music players, cameras, and more.
- Flash memory allows users to store data. It is sold in a variety of configurations, including compact flash, smart cards, and memory sticks, and they often connect to the computer through the USB port.

Components of I/O Subsystem



Channel

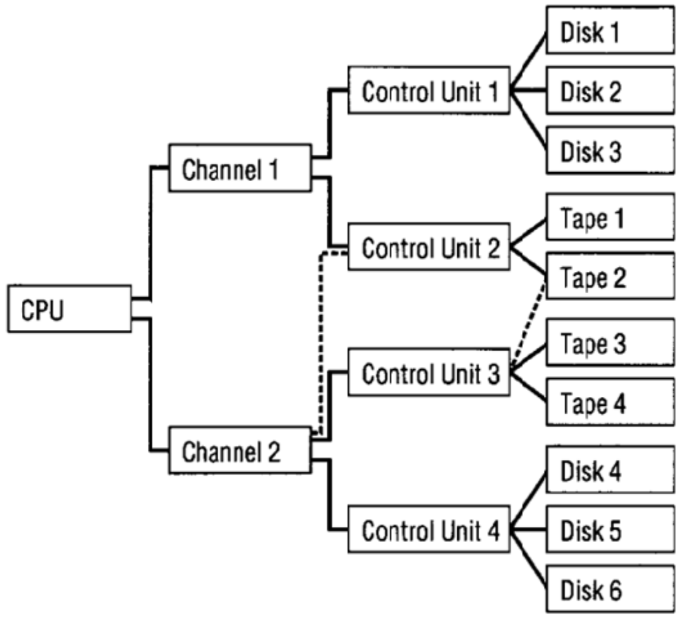
- **Channel** keep up with the I/O requests from the CPU and pass them down the line to the appropriate control unit.
- **I/O channels** are programmable units placed between the CPU and the control units.
- Their job is to synchronize the fast speed of the CPU with the slow speed of the I/O device, and they make it possible to overlap I/O operations with processor operations so the CPU and I/O can process concurrently.
- Channels use **an I/O channel program** which specifies the action to be performed by the devices and controls the transmission of data between main memory and the control units.

At the start of an I/O command, the information passed from the CPU to the channel is this: I/O command (READ, WRITE, REWIND, etc.), Channel number, Address of the physical record to be transferred (from or to secondary storage), Starting address of a memory buffer from which or into which the record is to be transferred

I/O Control Unit

The channel sends one signal for each function, and the **I/O control unit** interprets the signal. Although a control unit is sometimes part of the device, in most systems a single control unit is attached to several similar devices.

Because the channels are as fast as the CPU they work with, each channel can direct several control units by interleaving commands. A typical configuration might have one channel and up to eight control units, each of which communicates with up to eight I/O devices.



Additional flexibility can be built into the system by connecting more than one channel to a control unit or by connecting more than one control unit to a single device.
These multiple paths increase the reliability of the I/O subsystem by keeping communication lines open even if a component malfunctions.

COMMUNICATION AMONG DEVICES

The Device Manager relies on several auxiliary features to keep running efficiently under the demanding conditions of a busy computer system, and there are three problems that must be resolved:

- It needs to know which components are busy and which are free.
- It must be able to accommodate the requests that come in during heavy I/O traffic.
- It must accommodate the disparity of speeds between the CPU and the I/O devices.

Channel Status Word

- Each unit in the I/O subsystem can finish its operation independently from the others. Once a device has begun task the connection between the device and its control unit can be cut off so the control unit can initiate another I/O task with another device.

- The success of the operation depends on the system's ability to know when a device has completed an operation. This is done with a hardware flag that must be tested by the CPU.
- This flag is made up of three bits and resides in the **Channel Status Word (CSW)**, which is in a predefined location in main memory and contains information indicating the status of the channel.
- Each bit represents one of the components of the I/O subsystem, one each for the channel, control unit, and device.
- Each bit is changed from 0 to 1 to indicate that the unit has changed from free to busy.

Polling and Interrupt

There are two common ways to ensure that the entire path is free or busy. They are

1. polling
2. Interrupts

Polling

- Polling uses a special machine instruction to test the flag. The CPU periodically tests the channel status bit (in the CSW). If the channel is still busy, the CPU performs some other processing task until the test shows that the channel is free; then the channel perform the I/O operation.
- The major issue with this scheme is determining how often the flag should be polled.
- If polling is done too frequently the CPU wastes time testing the flag just to find out that the channel is still busy.
- If polling is done too seldom, the channel could sit idle for long periods of time.

Interrupts

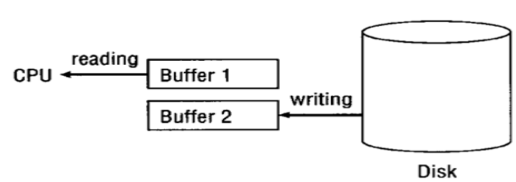
- The use of **interrupts** is a more efficient way to test the flag.
- Instead of having the CPU test the flag, a hardware mechanism does the test as part of every machine instruction executed by the CPU.
- If the channel is busy, the flag is set so that execution of the current sequence of instructions is automatically interrupted and control is transferred to the interrupt handler, which is part of the operating system and resides in a predefined location in memory.
- The interrupt handler's job is to determine the best course of action i.e., must find out which unit sent the signal, analyze its status, and restart it when appropriate with the next operation, and finally return control to the interrupted process.

Direct Memory Access

- Direct memory access (DMA) is an I/O technique that allows a control unit to directly access main memory. This means that once reading or writing has begun, the remainder of the data can be transferred to and from memory without CPU intervention.
- To activate this process, the CPU sends enough information such as the type of operation (read or write), the unit number of the I/O device needed, the location in memory where data is to be read from or written to, and the amount of data (bytes or words) to be transferred to the DMA control unit to initiate the transfer of data.
- The CPU then can go on to another task while the control unit completes the transfer independently.
- DMA Controller sends an interrupt to the CPU to indicate that the operation is completed. This mode of data transfer is used for high-speed devices such as disks.
- Without DMA, the CPU is responsible for the physical movement of data between main memory and the device which is a time-consuming task that results in significant over-head and decreased CPU utilization.

Buffers

- **Buffers** are used extensively to better synchronize the movement of data between the relatively slow I/O devices and the very fast CPU.
- Buffers are temporary storage areas residing in three convenient locations throughout the system: main memory, channels, and control units.
- They're used to store data read from an input device before it's needed by the processor and to store data that will be written to an output device.
- To minimize the idle time for devices to maximize their throughput, the technique of **double buffering** is used. In this system, two buffers are present in main memory, channels, and control units.
- The objective is to have a record ready to be transferred to or from memory at any time to avoid any possible delay that might be caused by waiting for a buffer to fill up with data.



Management of I/O Requests

The Device Manager divides its task into three parts with each one handled by a specific software component of the I/O subsystem.

- The I/O traffic controller watches the status of all devices, control units, and channels.
- The I/O scheduler implements the policies that determine the allocation of, and access to, the devices, control units, and channels.
- The I/O device handler performs the actual transfer of data and processes the device interrupts.

I/O traffic controller

The **I/O traffic controller** monitors the status of every device, control unit, and channel. The traffic controller has three main tasks:

- It must determine if there's at least one path available;
- If there's more than one path available, it must determine which to select;
- If the paths are all busy, it must determine when one will become available.

It's a job that becomes more complex as the number of units in the I/O subsystem increases and as the number of paths between these units increases.

To do all this, the traffic controller maintains a database containing the status and connections for each unit in the I/O subsystem, grouped into Channel Control Blocks, Control Unit Control Blocks, and Device Control Blocks,

Channel Control Block	**Control Unit Control Block**	**Device Control Block**
• Channel identification	• Control unit identification	• Device identification
• Status	• Status	• Status
• List of control units connected to it	• List of channels connected to it	• List of control units connected to it
• List of processes waiting for it	• List of devices connected to it	• List of processes waiting for it
	• List of processes waiting for it	

The **I/O scheduler** performs the same job as the Process Scheduler allocates the devices, control units, and channels. Under heavy loads, when the number of requests is greater than the number of available paths, the I/O scheduler must decide which request to satisfy first. Most commonly I/O requests are not preempted. Once the channel program has started, it’s allowed to continue to completion even though I/O requests with higher priorities may have entered the queue.

The **I/O device handler** processes the I/O interrupts, handles error conditions, and pro-vides detailed scheduling algorithms, which are extremely device dependent. Each type of I/O device has its own device handler algorithm.

DEVICE HANDLER SEEK STRATEGIES

A **seek strategy** for the I/O device handler is the predetermined policy that the device handler uses to allocate access to the device among the many processes that may be waiting for it.

It determines the order in which the processes get the device and the goal is to keep seek time to a minimum. Most commonly used seek strategies are

- First-come, first-served (FCFS)
- Shortest seek time first (SSTF)
- SCAN and its variations LOOK, C-SCAN, and C-LOOK.

Every scheduling algorithm should do the following

- Minimize arm movement
- Minimize mean response time
- Minimize the variance in response time

FIRST COME -FIRST SERVE (FCFS)

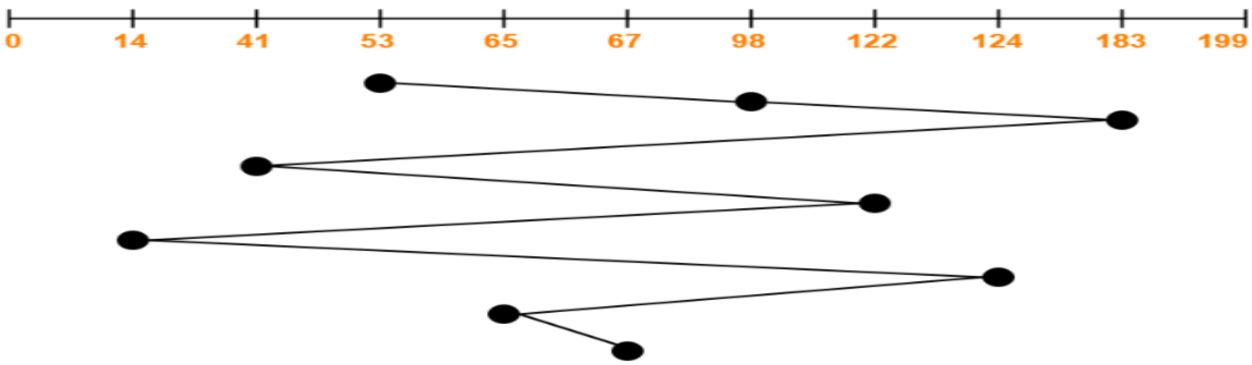
FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue. All incoming requests are placed at the end of the queue. Whatever number that is next in the queue will be the next number served. Using this algorithm doesn't provide the best results.

Problem-

Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65,

67. The FCFS scheduling algorithm is used. The head is initially at cylinder number 53. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is _______.

Solution-



Total head movements incurred while servicing these requests

= (98 – 53) + (183 – 98) + (183 – 41) + (122 – 41) + (122 – 14) + (124 – 14) + (124 – 65) + (67 – 65)

= 45 + 85 + 142 + 81 + 108 + 110 + 59 + 2

= 632

Advantages:

- Every request gets a fair chance
- No indefinite postponement

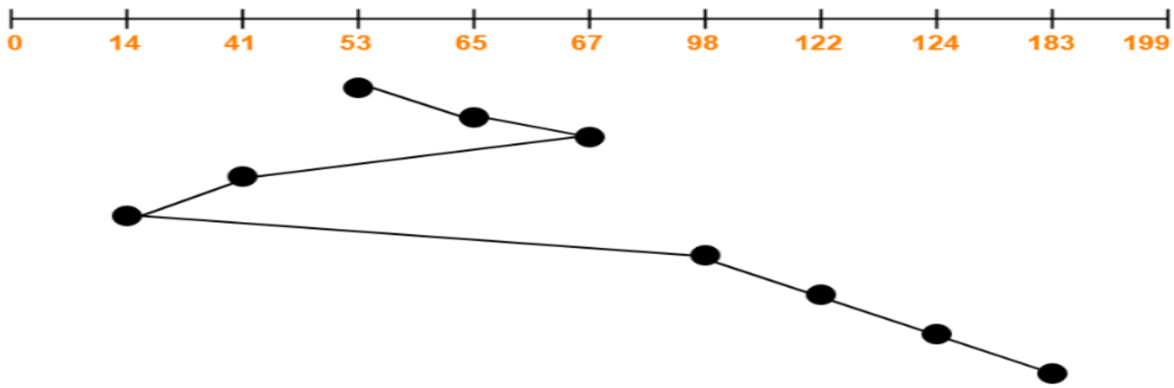
Disadvantages:

- Does not try to optimize seek time
- May not provide the best possible service

SHORTEST SEEK TIME FIRST (SSTF)

- In SSTF (Shortest Seek Time First), requests having shortest seek time are executed first.
- So, the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time.
- As a result, the request near the disk arm will get executed first.
- SSTF is certainly an improvement over FCFS as it decreases the average response time and increases the throughput of system.

Solution-



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (67 – 41) + (41 – 14) + (98 – 14) + (122 – 98) +

(124 – 122) + (183 – 124)

= 12 + 2 + 26 + 27 + 84 + 24 + 2 + 59

= 236

Advantages:

- Average Response Time decreases
- Throughput increases

Disadvantages:

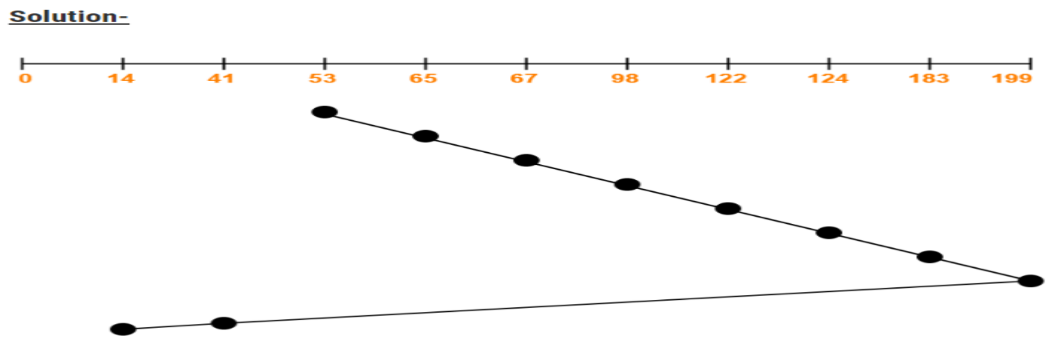
- Overhead to calculate seek time in advance
- Can cause Starvation for a request if it has higher seek time as compared to incoming requests
- High variance of response time as SSTF favor only some requests

There is a great chance that starvation would take place. The reason for this is if there were a lot of requests close to each other the other requests will never be handled since the distance will always be greater.

SCAN

- This algorithm scans all the cylinders of the disk back and forth.
- Head starts from one end of the disk and move towards the other end servicing all the requests in between.

- After reaching the other end, head reverses its direction and move towards the starting end servicing all the requests in between.
- The same process repeats.
- So, this algorithm works as an elevator and hence also known as **elevator algorithm.** As a result, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait.



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (199 – 183) + (199 – 41) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 158 + 27

= 331

Advantages-

- It is simple, easy to understand and implement.
- It does not lead to starvation.
- It provides low variance in response time and waiting time.

Disadvantages-

- It causes long waiting time for the cylinders just visited by the head.
- It causes the head to move till the end of the disk even if there are no requests to be serviced.

C-SCAN

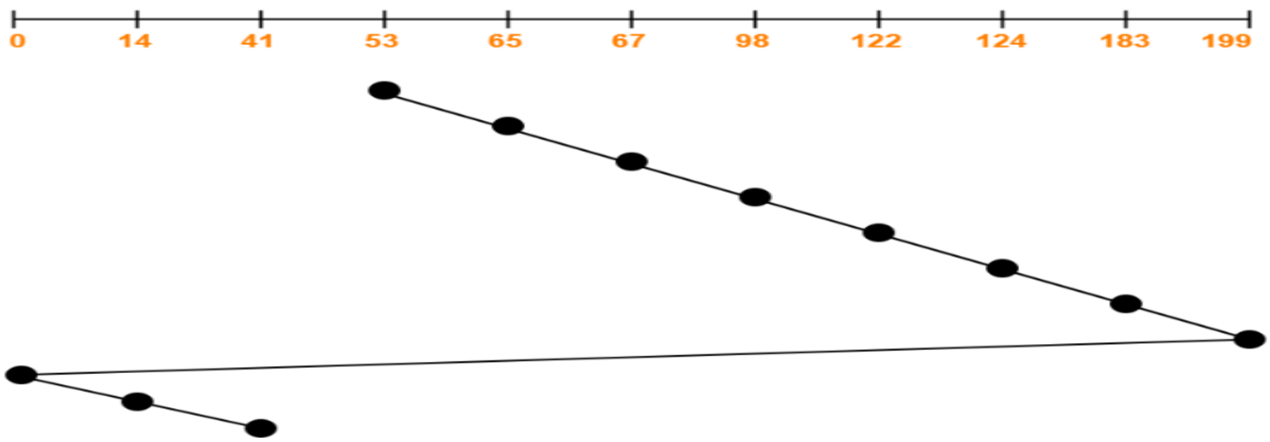
It begins its scan toward the nearest end and works all the way to the end of the system.

Once it hits the bottom or top it jumps to the other end and moves in the same direction.

Advantages:

- Provides more uniform wait time compared to SCAN

Solution-



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (199 – 183) + (199 – 0) + (14 – 0) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 199 + 14 + 27

= 386

LOOK Disk Scheduling Algorithm-

- LOOK Algorithm is an improved version of the **SCAN Algorithm**.
- Head starts from the first request at one end of the disk and moves towards the last request at the other end servicing all the requests in between.
- After reaching the last request at the other end, head reverses its direction.
- It then returns to the first request at the starting end servicing all the requests in between.
- The same process repeats.

NOTE-

The main difference between SCAN Algorithm and LOOK Algorithm is-

- SCAN Algorithm scans all the cylinders of the disk starting from one end to the other end even if there are no requests at the ends.

- LOOK Algorithm scans all the cylinders of the disk starting from the first request at one end to the last request at the other end.

Advantages-

It does not cause the head to move till the ends of the disk when there are no requests to be serviced.

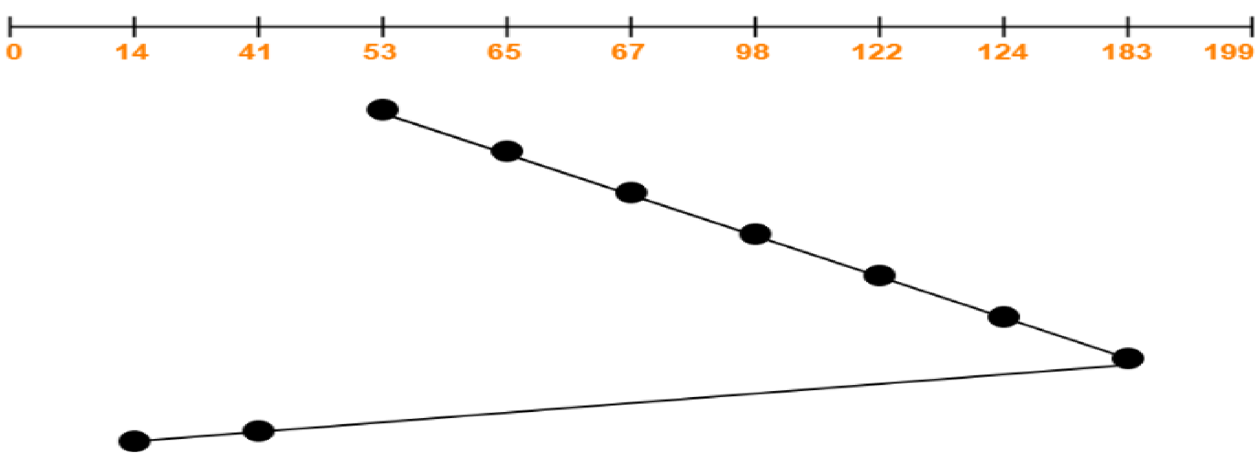
- It provides better performance as compared to SCAN Algorithm.
- It does not lead to starvation.
- It provides low variance in response time and waiting time.

Disadvantages-

There is an overhead of finding the end requests.

- It causes long waiting time for the cylinders just visited by the head.

Solution-



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) +

(183 – 41) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 142 + 27

= 299

C-LOOK Disk Scheduling Algorithm-

- Circular-LOOK Algorithm is an improved version of the **LOOK Algorithm**.
- Head starts from the first request at one end of the disk and moves towards the last request at the other end servicing all the requests in between.
- After reaching the last request at the other end, head reverses its direction.
- It then returns to the first request at the starting end without servicing any request in between.
- The same process repeats till all the requests have been processed.

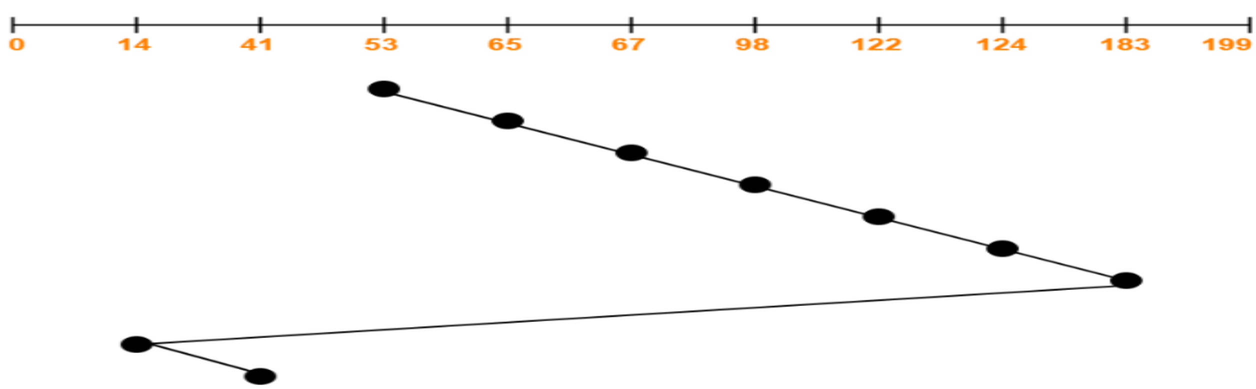
<u>Advantages-</u>

- It does not cause the head to move till the ends of the disk when there are no requests to be serviced.
- It reduces the waiting time for the cylinders just visited by the head.
- It provides better performance as compared to LOOK Algorithm.
- It does not lead to starvation.
- It provides low variance in response time and waiting time.

<u>Disadvantages-</u>

- There is an overhead of finding the end requests.

<u>Solution-</u>



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) +

(183 – 14) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 169 + 27

= 326

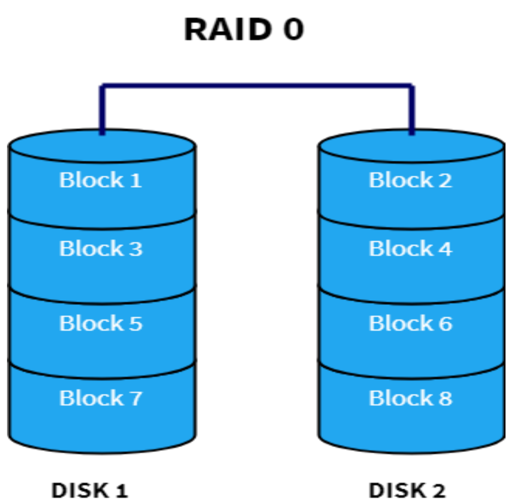
RAID (Redundant Arrays of Independent Disks)

- RAID means "Redundant Arrays of Independent Disks" or "Redundant Arrays of Inexpensive Disks".
- RAID is a storage virtualization technology which is used to organize multiple drives into various arrangements to meet certain goals like redundancy, speed and capacity.
- Is a technique which makes use of a combination of multiple disks instead of using a single disk for increased performance, data redundancy or both
- Data redundancy, although taking up extra space, adds to disk reliability. This means, in case of disk failure, if the same data is also backed up onto another disk, it is possible to retrieve the data and go on with the operation.
- RAID can be categorized into Software RAID and Hardware RAID. In software RAID, the memory architecture is managed by the operating system.
- In case of hardware RAID, there is a dedicated controller and processor present inside the disks that manage the memory.

RAID Level	Error Correction Method	I/O Request Rate	Data Transfer Rate
0	None	Excellent	Excellent
1	Mirroring	Read: Good Write: Fair	Read: Fair Write: Fair
2	Hamming code	Poor	Excellent
3	Word parity	Poor	Excellent
4	Strip parity	Read: Excellent Write: Fair	Read: Fair Write: Poor
5	Distributed strip parity	Read: Excellent Write: Fair	Read: Fair Write: Poor
6	Distributed strip parity and independent data check	Read: Excellent Write: Poor	Read: Fair Write: Poor

RAID 0

- RAID 0 is based on data striping.
- **Striping** is writing data across a number of disks in parallel, which speeds read/write performance.
- A stream of data is divided into multiple segments or blocks and each of those blocks is stored on different disks.
- So, when the system wants to read that data, it can do so simultaneously from all the disks and join them together to reconstruct the entire data stream.
- The benefit of this is that the speed increases drastically for read and write operations.
- It is great for situations where performance is a priority over other aspects.
- Also, the total capacity of the entire volume is the sum of the capacities of the individual disks.



Advantages:

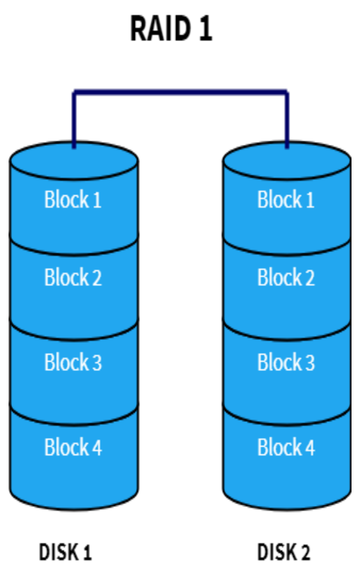
- Performance boost for read and write operations
- Space is not wasted as the entire volume of the individual disks are used up to store unique data

Disadvantages

- There is no redundancy/duplication of data. If one of the disks fails, the entire data is lost.

RAID 1

- RAID 1 uses the concept of data mirroring.
- Data is mirrored or cloned to an identical set of disks so that if one of the disks fails, the other one can be used.
- It also improves read performance since different blocks of data can be accessed from all the disks simultaneously.
- But unlike RAID 0, write performance is reduced since all the drives must be updated whenever new data is written. Another disadvantage is that space is wasted to duplicate the data thereby increasing the cost to storage ratio.



Advantages

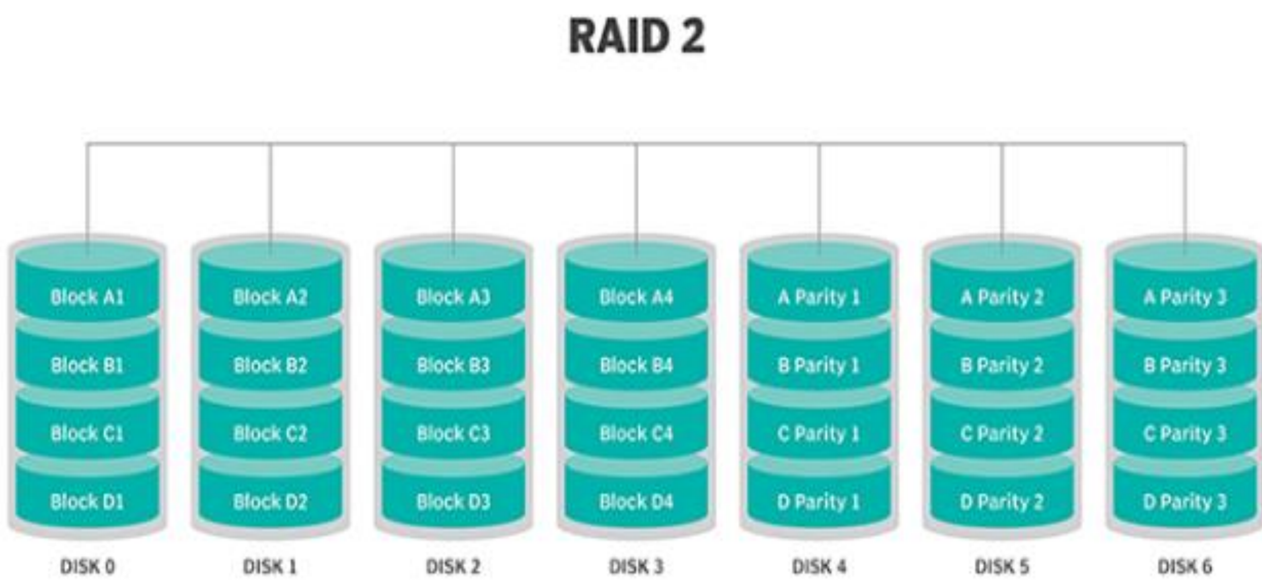
- Data can be recovered in case of disk failure
- Increased performance for read operation

Disadvantages

- Slow write performance
- Space is wasted by duplicating data which increases the cost per unit memory

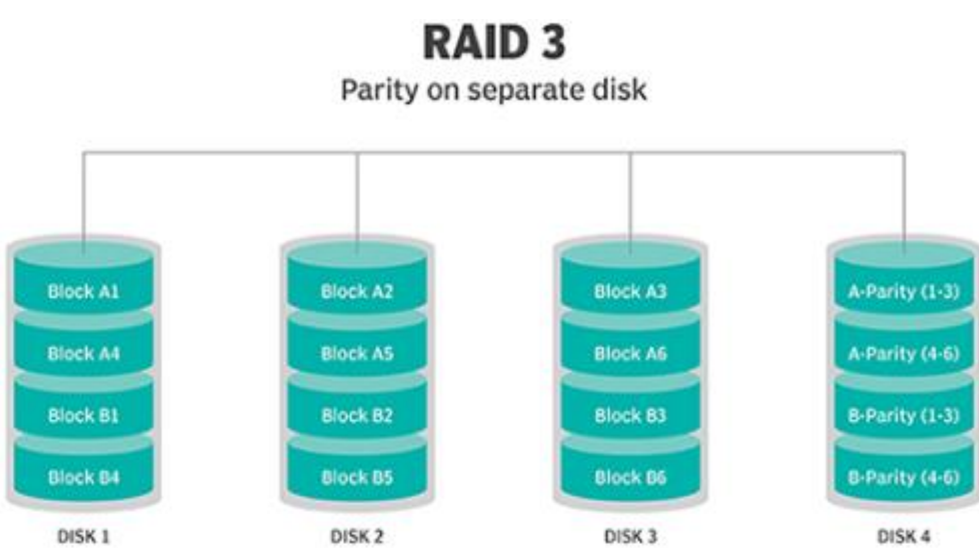
RAID 2

This configuration uses striping across disks, with some disks storing error checking and correcting (ECC) information. RAID 2 also uses a dedicated Hamming code parity; a linear form of error correction code. RAID 2 has no advantage over RAID 3 and is no longer used.



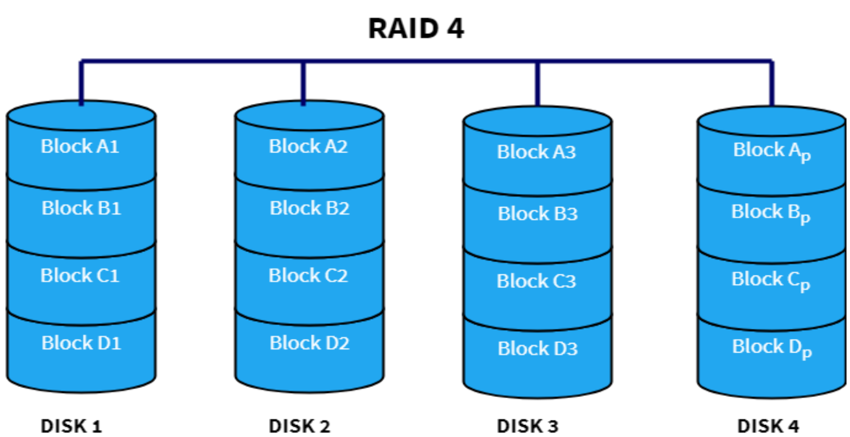
RAID 3

- This technique uses striping and dedicates one drive to storing parity information. The embedded ECC information is used to detect errors.
- Data recovery is accomplished by calculating the exclusive information recorded on the other drives. Since an I/O operation addresses all the drives at the same time, RAID 3 cannot overlap I/O.
- RAID 3 is best for single-user systems with long record applications.



RAID 4

- RAID 4 stripes the data across multiple disks just like RAID 0.
- It also stores parity information of all the disks in a separate dedicated disk to achieve redundancy.
- In the diagram below, Disk 4 serves as the parity disk having parity blocks A_p, B_p, C_p and D_p.
- Parity error checking is where redundancy information is calculated for each piece of data stored. If a drive fails, the missing data can be reconstructed from the remaining data and the parity data.
- Space is more efficiently used here when compared to RAID 1 since parity information uses way less space than mirroring the disk.
- Error checking tends to slow the system because data from several locations must be read and compared.



Advantages

- Efficient data redundancy in terms of cost per unit memory
- Performance boost for read operations due to data stripping

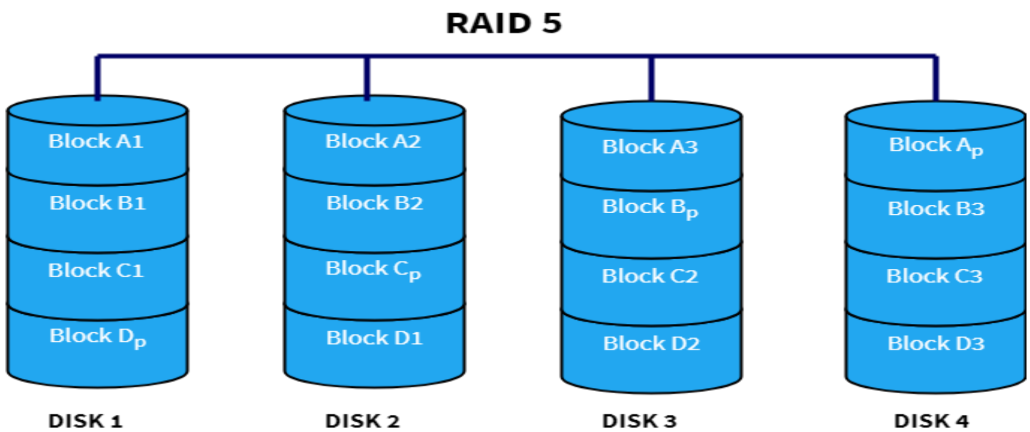
Disadvantages

- Write operation is slow
- If the dedicated parity disk fails, data redundancy is lost

RAID 5

- RAID 5 is very similar to RAID 4, but here the parity information is distributed over all the disks instead of storing them in a dedicated disk.

- This has two benefits: there is no more a bottleneck as the parity stress evens out by using all the disks to store parity information
- There is no possibility of losing data redundancy since one disk does not store all the parity information.



Advantages

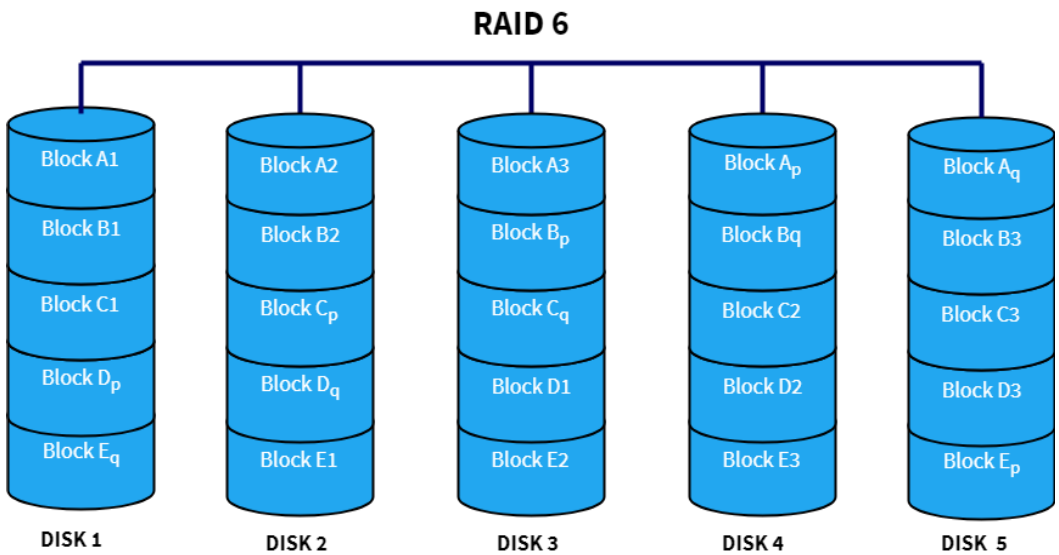
- All the advantages of RAID 4 plus increased write speed and better data redundancy

Disadvantages

- Can only handle up to a single disk failure

RAID 6

- RAID 6 uses double parity blocks to achieve better data redundancy than RAID 5.
- This increases the fault tolerance for up to two drive failures in the array.
- Each disk has two parity blocks which are stored on different disks across the array.
- RAID 6 is a very practical infrastructure for maintaining high availability systems.



Advantages

- Better data redundancy. Can handle upto 2 failed drives

Disadvantages

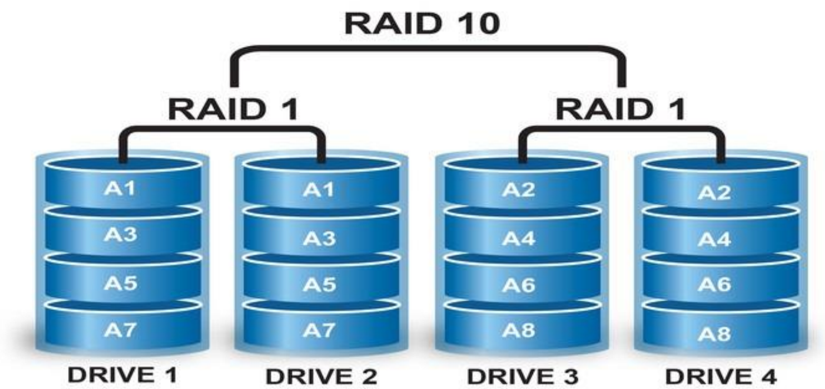
- Large parity overhead

Nested RAID Levels:

RAID 10 (RAID 1+0)

RAID 10 combines both RAID 1 and RAID 0 by layering them in opposite order. Sometimes, it is also called as "nested" or "hybrid" RAID. This is a "best of both worlds approach", because it has the fast performance of RAID 0 and the redundancy of RAID 1. In this setup, multiple RAID 1 blocks are connected with each other to make it like RAID 0. It is used in cases where huge disk performance (greater than RAID 5 or 6) along with redundancy is required.

RAID Level	Combinations
01 (or 0+1)	A Level 1 system consisting of multiple Level 0 systems
10 (or 1+0)	A Level 0 system consisting of multiple Level 1 systems
03 (or 0+3)	A Level 3 system consisting of multiple Level 0 systems
30 (or 3+0)	A Level 0 system consisting of multiple Level 3 systems
50 (or 5+0)	A Level 0 system consisting of multiple Level 5 systems
60 (or 6+0)	A Level 0 system consisting of multiple Level 6 systems



Advantages

- Very fast performance
- Redundancy and fault tolerance

Disadvantages

- Cost per unit memory is high since data is mirrored