26/3/24

ASSIGNMENT 2

Unit - 4,5 & 6 Memory Management, Mass Storage & File Management.

Q.1 Calculate how many page faults will occur for the following algorithms applied on given ~~refrence~~ reference string with three-pages frames. 1 2 3 2 1 5 2 1 6 2 5 6 3 1 3 6 1 2 4 3, ~ Reference String

i) FIFO

S.No	1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
F1	1	1	1	1	1	5	5	5	5	2	2	2	2	1	1	1	1	1	4	4
F2		2	2	2	2	2	2	1	1	1	5	5	5	5	5	6	6	6	6	3
F3			3	3	3	3	3	3	6	6	6	6	3	3	3	3	3	2	2	2
	F	F	F	H	H	F	H	F	F	F	F	H	F	F	H	F	H	F	F	F
	-	-	-			-		-	-	-	-		-	-		-		-	-	-

∴ Total Page Faults = 14.

ii) Optimal Page Replacement

S.NO	1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
F1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	2	2	2
F2		2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	4	4
F3			3	3	3	5	5	5	5	5	5	5	3	3	3	3	3	3	3	3
	F	F	F	H	H	F	H	H	F	H	H	H	F	F	H	H	H	F	F	H
	-	-	-			-			-				-	-				-	-	

∴ Total Page Faults = 9

iii) LRU Page Replacement

S.No	1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
F1	1	1	1	1	1	1	1	1	1	1	5	5	5	1	1	1	1	1	1	3
F2		2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	2	2	2
F3			3	3	3	5	5	5	6	6	6	6	6	6	6	6	6	6	4	4
	F	F	F	H	H	F	H	H	F	H	F	H	F	F	H	H	H	F	F	F
	-	-	-			-			-		-		-	-				-	-	-

∴ Total Page Faults = 11

Q.2 Explain how segmentation is achieved using hardware.

→ Segmentation is a memory management technique that allows the memory to be divided logically into segments of variable lengths, where each segment represents a logical unit such as a procedure, function, method or object.

i) Segment Descriptor Tables.

Hardware maintains segment descriptor tables in memory which contains information about each segment. They are typically organized in a hierarchical manner, with global descriptor tables (GDT) and local descriptor tables (LDT)

ii) Segment Registers.

Segment Registers holds segment selectors / indexes pointing to segment descriptor entries. It includes Code Segment (CS), Data segment (DS), Stack segment (SS) and Extra Segment (ES).

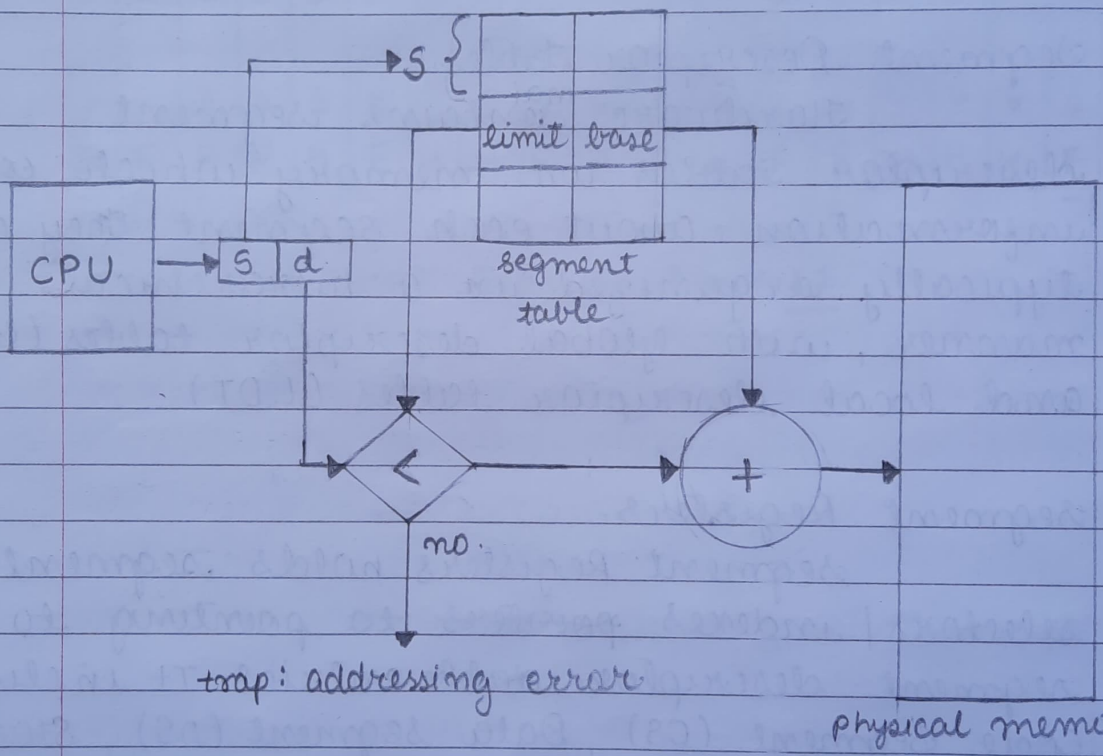
iii) Segment Selector

When a program accesses memory, it specifies a segment selector, which is an index into the segment descriptor tables. It provides necessary information to locate the segment descriptor corresponding

to the desired segment.

iii) Address Translation

The Hardware translates the logical address into a physical address and uses the segment selector to index into the SDT to retrieve the base address and limit of the corresponding segment.



Q.3 Explain link list method and bit map method for dynamic memory management.

→ Dynamic Memory Management is a ~~is~~ crucial aspect where memory allocation and ~~data~~ deallocation are frequent tasks.

i) Linked List Methods.

→ Here, the memory blocks are managed

using a linked list data structure, where each memory block is represented by a node in the linked list.

~4 Each node contains information about the memory block and a pointer to the next node.

~4 Initially, the entire memory is considered as one large free block. As memory allocations are made, this free block is split into smaller blocks and corresponding nodes are created and inserted in the list.

ii) Bitmap Methods

~4 Here, a bitmap is used to represent the allocation status of each memory block in a fixed-size memory pool.

~4 Each bit~~ma~~ in the bitmap corresponds to a memory block and its value indicates whether the corresponding block is allocated or free. Example, a '0' bit may represent a free block, while a '1' bit may represent an allocated block.

~4 Initially, the entire bitmap is initialized to indicate that all memory blocks are free.

~4 When memory is allocated, the bitmap is updated to reflect the allocation status of the corresponding block.

Q.4 write in detail about memory ~~partitioning~~ partitioning with fixed and dynamic

partitioning with example.

→ Memory Partitioning is used to divide the available memory into multiple partitions to accomodate multiple processes concurrently.

i) Fixed Partitioning

Here, the available memory is divided into fixed-size partitions or regions before the execution of any process. Each partition is allocated to a single process and the size of ~~eat~~ each partition remains constant throughout the execution.

Processes are assigned to partitions based on their size and a process cannot exceed the size of its assigned partitions.

It is relatively simple to implement but may lead to internal Fragmentation, where a partition may not be fully utilized, resulting in wasted memory.

eg: Consider a system with 1000 KB of memory divided into four fixed-size partitions and four processes arrives for execution:

- Partition-1 : 200 KB ← PA : 150 KB
- Partition-2 : 300 KB ← PB : 250 KB
- Partition-3 : 400 KB ← PC : 350 KB
- Partition-4 : 100 KB ← PD : 200 KB

Here, Partition 2 and 4 will have internal fragmentation.

ii) Dynamic Partitioning

Here, it allows for more flexible memory allocation by dividing the available memory into variable-sized partitions based on the size of processes.

When a process arrives for execution, it is allocated precisely according to the size of it, so it eliminates internal fragmentation.

However, it may suffer from external fragmentation, where small blocks of free memory sp become scattered throughout the memory space, making it challenging to allocate contiguous memory for larger processes.

eg: Consider a system of 1000 KB of memory initially free and processes arrive for execution so,
- Process A (200 KB) arrives first and is allocated memory from 0 KB to 200 KB. and so on from 201 KB

Q.5 What is physical address and logical address? How the mapping is done is between them.

—> i) Physical Address

It refers to the actual location of a memory cell in the computer's physical memory (RAM). It represents a unique location in the hardware memory module. They are used by the memory management unit (MMU) and memory hardware to access

and manipulate data directly in the physically memory.

The physical address space is the total range of addresses that the hardware to can reference directly. The size of it the physical address space is determined by the architecture and limitations of the underlying hardware.

ii) Logical Address

It is also known as a virtual address, is an address generated by the CPU during program execution. It represents a location in the logical address space of a process.

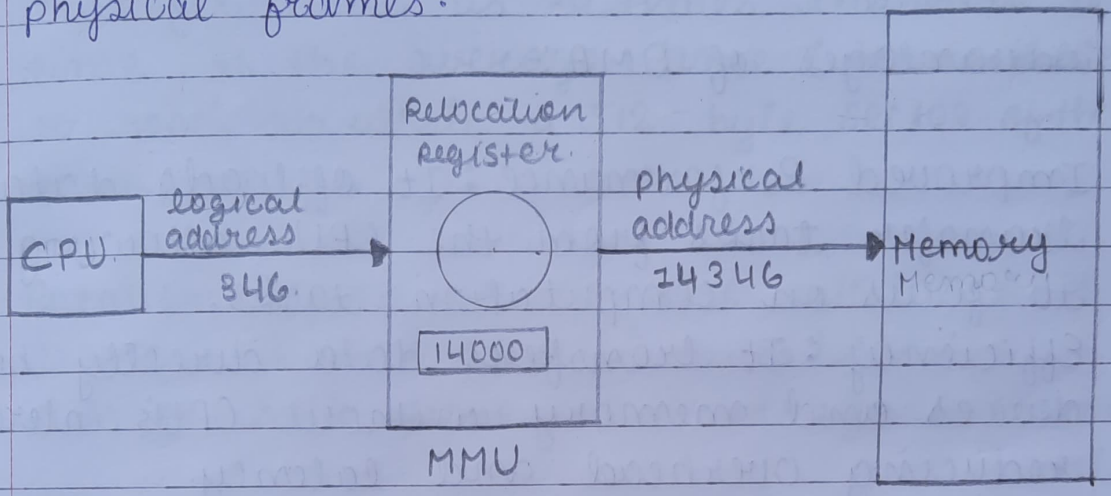
They are used by programs and the operating systems to refer ~~to process~~ to memory locations without concerning themselves with the actual physical memory layout.

It is the total range of addresses that a process can use. It is typically larger than the physical addresses space and is managed by the operating systems.

They are translated into physical addresses by the memory management unit (MMU) before accessing the actual physical memory.

-4 The mapping between logical and physical

addresses is done through a process known as Address Translation. The OS maintains a data structure called the page table for each process and it contains mappings between logical pages and physical frames.



Q.6 Explain in detail about DMA.

→ DMA (Direct Memory Access) allows certain hardware ~~components~~ subsystems to access the system's memory for reading from or writing to it independently of the central processing unit. DMA is commonly used for high-speed data transfer between peripheral devices and memory without involving the CPU.

- Components of DMA :-

i) DMA Controller : Responsible for managing DMA transfers between peripherals and memory without CPU intervention.

ii) DMA Channels: Allows simultaneous data transfers between different peripherals and memory.

iii) Memory Buffer: Regions of system memory allocated to store data being transferred via DMA.

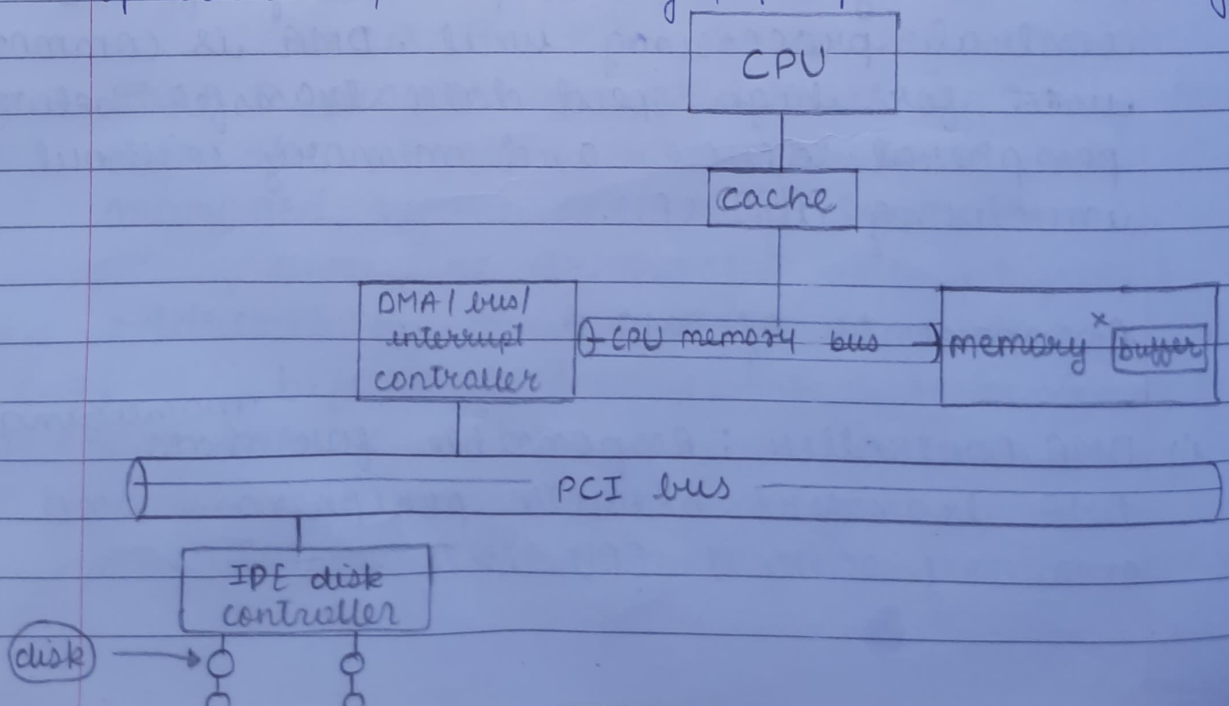
- Advantages of DMA:-

i) Improved Performance: It offloads data transfer tasks from the CPU, allowing it to focus on computation tasks.

ii) Efficiency: It transfers data directly between devices and memory without CPU's interactions, reducing overhead and latency.

iii) Asynchronous Operation: Allows for asynchronous data transfer, enabling concurrent data transfer and computation.

~4 DMA finds widespread applications like Disk I/O operations, Networking, Graphics Processing.



Q.7 Consider a typical disk that rotates at 15000 rotations per minute (RPM) and has a transfer rate of 50×10^6 bytes / sec. If the average seek time of the disk is twice the average rotational delay and the controller's transfer time is 10 times the disk transfer time, the average time (in milliseconds) to read or write a 512-byte sector of the disk is:

→ Given,

- Rotation Speed of the disk = 15000 RPM
- Transfer Rate = 50×10^6 bytes / sec.
- Average Seek Time = 2 × Average Rotational delay.
- Controller's transfer time = 10 × Disk transfer time.

∴ Time taken for one full rotation

$= \frac{60}{15000}$ sec

$= 0.004$ sec

$= 4$ ms

∴ Average Rotational Delay

$= \frac{1}{2} \times$ time taken for one full rotation

$= \frac{1}{2} \times 4$

$= 2$ ms

∴ Disk Transfer Time
= Time taken to read/write 512 bytes
= $\frac{512 \text{ bytes}}{50 \times 10^6 \text{ bytes/sec}}$

= 10.24×10^{-6} sec
= 0.01024 ms

∴ Controller's Transfer Time
= 10 × Disk transfer time
= 10 × 0.01024 ms
= 0.1024 ms

∴ Avg time to read/write 512 bytes
= Avg seek time + Avg rotational delay + Disk Transfer time + Controller's transfer time

= 4 + 2 + 0.01024 + 0.1024 ms
= 6.11 ms

Q.8 Explain techniques for performing I/O and describe external devices that engage in I/O with computer systems.

→ Input/Output (I/O) operations are essential for computer systems to interact with external devices and networking devices.

→ Techniques for Performing I/O :-

i) Programmed I/O (PIO)

Here, the CPU directly controls data transfer between the external device and memory. The CPU issues commands to the device, waits for the device to complete the operation and then reads or writes data.

ii) Interrupt-Driven (I/O)

Here, the CPU initializes the I/O operation and then proceeds with other tasks. When the I/O devices completes the operation, it interrupts the CPU to signal completion.

iii) Direct Memory Access (DMA)

Here, it allows external devices to transfer data directly to/from memory without CPU intervention. The CPU sets up the DMA controller with transfer parameters and initiates the transfer.

→ External Devices Engaging in I/O:

i) Storage Devices

Hard Disk Drives (HDDs), Solid-State drives (SSDs) and optical drive (CD/DVD/Blu-ray) are common storage devices which allows engage in I/O operation to read/write data from/to the computer's memory, facilitating data storage & retrieval

ii) Input Devices

Keyboards, mice, touchscreens, scanners and ~~camera~~ cameras, which send data to the computer system for processing via I/O operations.

iii) Output Devices.

Monitors, printers, speakers and projectors, they receive data from the computer system and present it to users through visual, audio or printed output.

iv) Networking Devices

Network Interface Cards (NICs), routers, switches and modems, engages in I/O operations to send and receive data over computer networks.

Q.9 What is the need of file system? Explain various operations associated with files.

- The file system is a crucial component responsible for managing files and directories on storage devices which provides a structured way to structured way of to organize, store, retrieve and manage data efficiently.

i) Data Organization.

It ~~provid~~ organizes data into files and directories, providing a hierarchical structure that makes it easier for users and applications to locate and access relevant data.

ii) Data Persistance.

It ensures that data persists even after the computer system is powered off. It manages the storage of files on non-volatile storage devices, ensuring that data remains intact across system reboots.

iii) Data Sharing

It facilitate sharing of data among multiple users and applications. They provide mechanisms for controlling access permissions to files and directories, allowing users to share or restrict access to data as needed.

* Data integrity, Data Protection and Optimized

Access are also main needs for a file system.

→ Various Operations Associated with Files:

i) Creating Files: Operation of creating a new file involves allocating space on the storage device and assigning a unique identifier to the file.

ii) Opening Files: Opening a file allows a process to access the contents of the file for reading, writing or both.

iii) Closing Files: Closing a file releases any resources associated with the file and updates the file's metadata to reflect that it is no longer in use by the process.

iv) Reading from Files: Reading from a file involves retrieving data from the file and transferring it to the requesting process's memory space.

v) Writing to Files: Writing to a file involves storing data provided by a process to the file on the storage devices.

vi) Seeking within Files: Seeking within a file involves moving the file pointer to a specified position within the file, allowing

processes to read from or write to different parts of the file.

vii) Deleting Files: Deleting a file removes it from the file system and frees up the space occupied by the file on the storage device.

Q.10 Describe files sharing system and describe some of the difficulties while involving users to share file.

→ A file sharing system allows multiple users to access and collaborate on files stored on a shared storage devices or server. It enables users to share files, documents and other resources with each other over a network, facilitating collaboration, communication and data exchange.

i) Multiple users: It supports multiple users accessing and working on the same set of files.

ii) Client - Server Model: Here, a server hosts the files and clients connect to the server to access and modify these files. The server controls access permissions and ensure data integrity

iii) Remote File Systems: With the advent

of networking technologies, remote file systems have become prevalent. Users can access files stored on servers located in different physical locations.

→ Difficulties in File Sharing :-

i) Conflict Resolution : When multiple users attempt to modify the same file simultaneously, conflict can arise.

ii) Network Interruptions : File sharing over networks introduces ~~vvvv~~ vulnerabilities to network interruptions. Server failures can disrupt ~~access~~ access to shared files and require recovery mechanisms.

iii) Data Integrity : Maintaining data integrity, especially in scenarios where multiple users are modifying files concurrently, requires careful design and implementation of file systems.

iv) Performance : As the number of users and the size of shared files increase, file sharing systems must handle large amounts of data efficiently.

v) Access Control : Balancing ease of access with security is a challenge. Administrators

must carefully manage access permissions to prevent unauthorized users from accessing sensitive data.

Q.11 Describe the file system architecture.

—4 The file system architecture typically consists of several layers that work together to manage files and provide access to data stored on storage devices.

i) Application Programs

At the topmost layer are application programs that interact with files. These programs make requests to the file system to perform operations such as reading, writing and deleting files. These programs include word processors, spreadsheets, media players that users use to create, modify and manipulate files.

iii) File - Organization Module.

The file organization module sits between the application programs and the basic file system. It provides a logical interface for organizing files and accessing data. This module defines how files are structured, named and organized within the file systems. It also manages directory structures and metadata associated with files.

iv)

Basic File System.

The basic file system layer is responsible for managing the physical storage of files on storage devices. It handles tasks such as allocating and deallocating storage space, managing file metadata and data blocks and performing low-level I/O operations.

v)

I/O Control

The I/O control layer is responsible for controlling the interaction between the file system and I/O devices. It manages device drivers and handles requests to read from and write to storage devices. This layer ensures efficient and reliable communication between the file system and storage hardware.

vi)

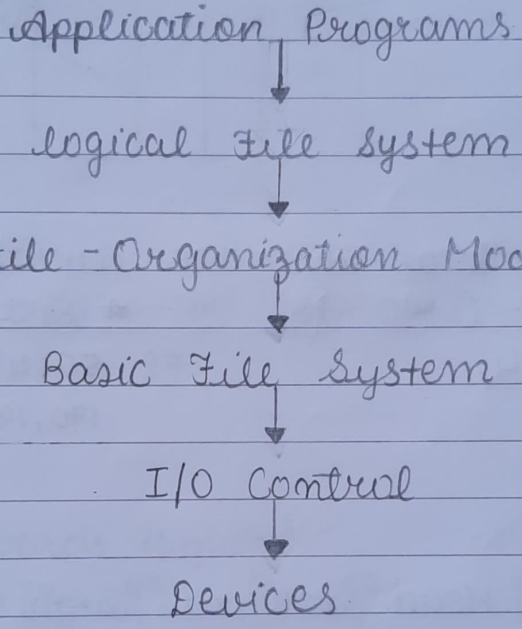
Devices

At the lowest layer are physical storage devices. These devices store the actual data and provide mechanisms for reading and writing data blocks. The devices layer interacts directly with the hardware components of the computer system.

ii) Logical File System

Some file systems include an additional layer known as the logical file

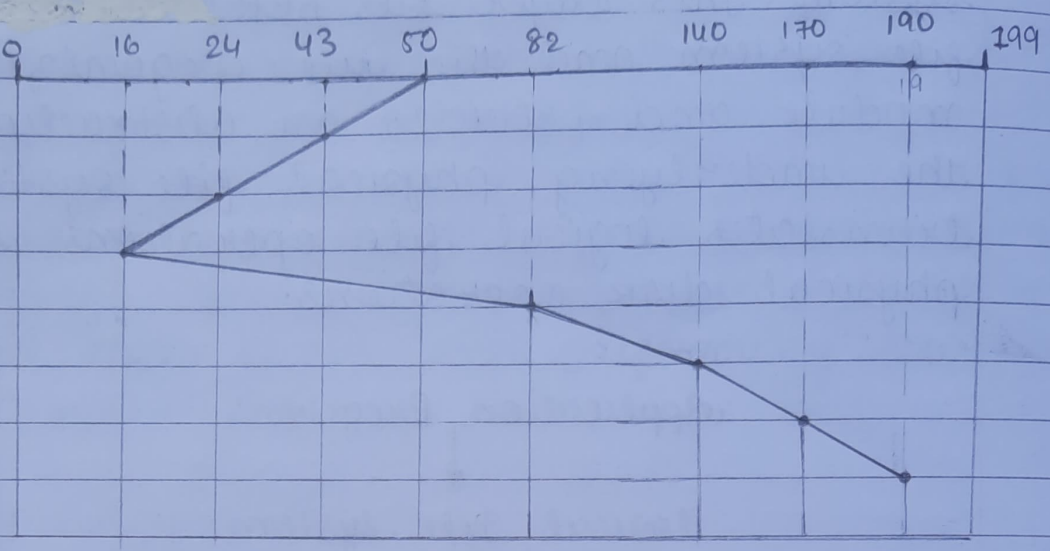
system. This layer sits between the basic file system and the file-organization module and provides an abstraction of the underlying physical file system. It translates logical file operations into physical disk operations.



Q.12 Suppose a disk drive has 200 cylinders. The drive is currently serving request at 50. The queue of pending request in FIFO order is 82, 170, 43, 140, 24, 16, 190. Starting from the current head position what is the total distance that the disk arm moves to satisfy the entire pending request for each of the following disk Scheduling algorithm.

i) SSTF

—ϕ



Total distance covered by the disk.

= (50 - 16) + (190 - 16)

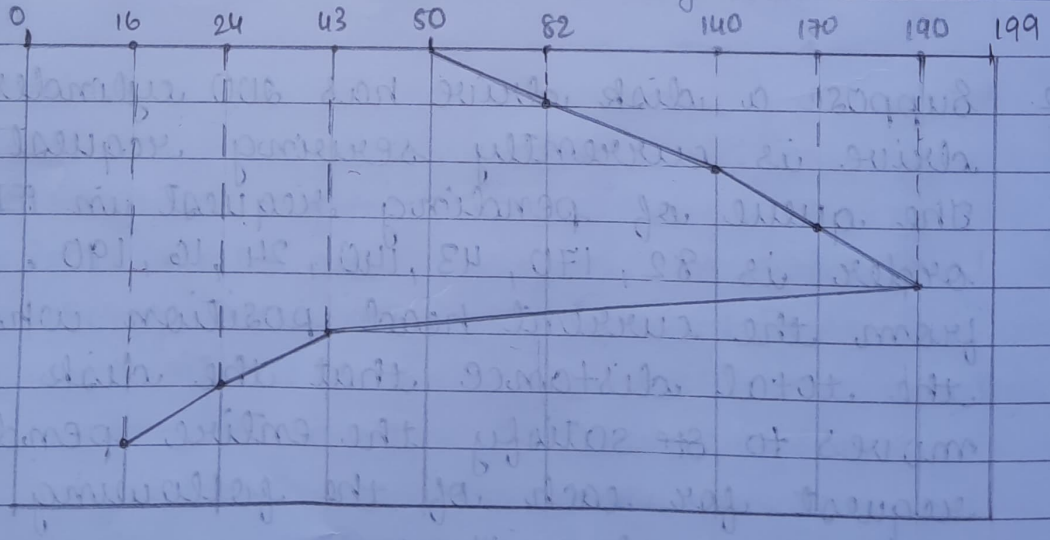
= 34 + 174.

= 208

★ Disk Movement:
50, 43, 24, 16, 82, 140, 170, 190

ii) LOOK

—ϕ i) From R/W Head towards higher value.



∴ Total distance covered by the disk

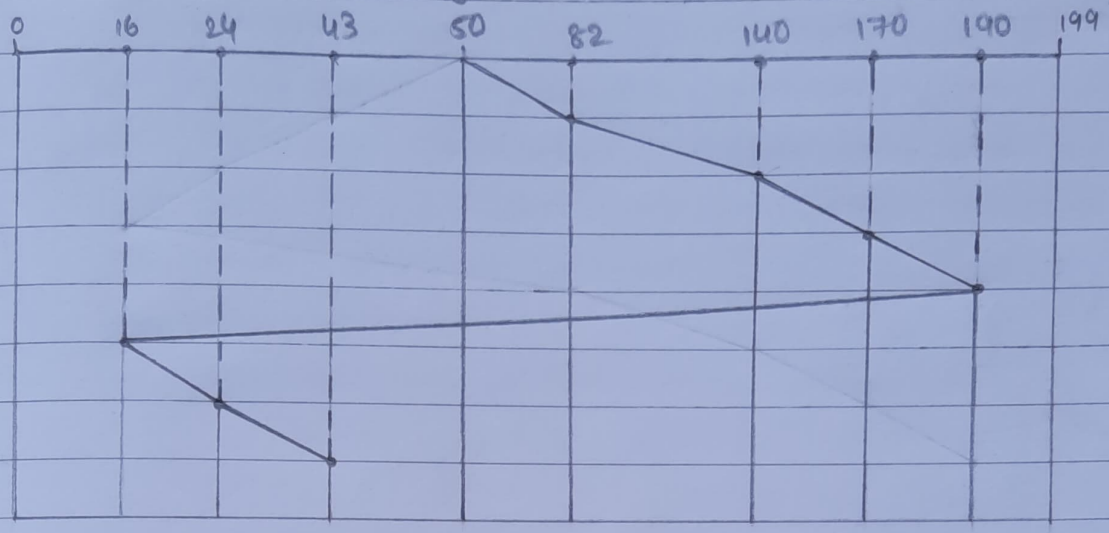
= (190 - 50) + (190 - 16)

= 140 + 174

= 314

★ Disk Movement
50, 82, 140, 170, 190, 43, 24, 16.

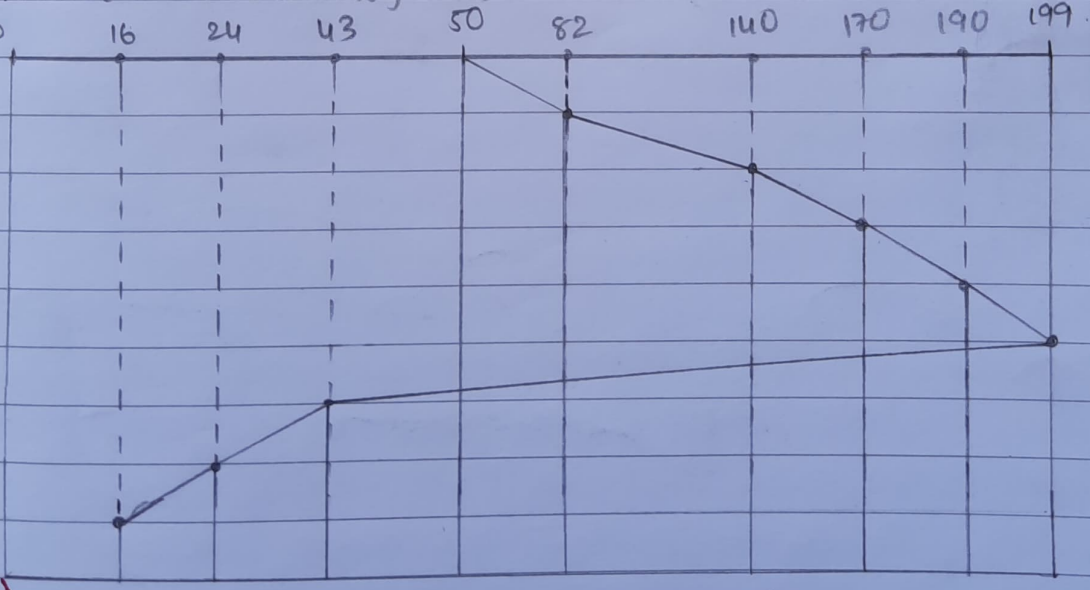
iii) C LOOK (Towards Higher)



∴ Total distance covered by the disk
= (190 - 50) + (190 - 16) + (43 - 16)
= 140 + 174 + 27.
= 341

★ Disk Movement:
50, 82, 140, 170, 190,
16, 24, 43.

iv) SCAN (Towards Higher)



Total distance covered by the disk
= (199 - 50) + (199 - 16)
= 332

★ Disk Movement:
50, 82, 140, 170, 190, 199,
43, 24, 16