

## Unit - IV (Device and Storage Management)

### # Storage Management

Storage management is defined as the management of data storage equipment that are used to store computer generated data. It is a tool or set of process used by administrator to keep your data and storage equipment safe.

### \* Key attributes of Storage Management

- Performance
- Reliability
- Recoverability
- Capacity

### \* Features of Storage Management

- Storage management is a process that is used to optimize the use of storage devices.
- Storage management must be allocated as a resource.
- Storage management is a basic system component of computer.
- It is used to improve the performance of data storage resources.

SujAL

### \* Advantage of Storage Management

- More secure data
- Better performing app.
- Better user experience.
- More efficient operation.

### \* Disadvantage of Storage Management

- Data in the cloud storage can lost with the malware attack and natural disaster.
- All data stored in multiple data storage devices in data centre will corrupted.

### \* Limitation

- Limited physical storage
- Complexity of storage management
- Security issue
- High cost
- Backup and recovery

## (#) Device Management

Device management is a process of controlling the Input / Output devices and other supporting and control unit.

→ The fundamental of I/O devices that need to be managed is divided into 3 categories.

### (1) Boot Devices

It stores data into fixed - sized blocks, each with its own unique address.

### (2) Character Devices

It sends or receives a continuous stream of character, none of which can be addressed individually. Ex - Keyboard, Printer

### (3) Network Device

It is used for data packet transmission.

### \* Feature of Device Management

- O/S interact with the device controller through device drivers.
- D/M is responsible for implementing APIs.
- Device controller used in D/M operation consist of three registers : Command, status and data.

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

## \* Types of Devices

### → Dedicated Devices

These devices are designed to allocate a single task at a time. Ex - Keyboard, monitor.

The disadvantage of this device is it is less efficient than other device because you can perform one task at a time.

### → Shared Devices

Shared devices are assigned for multiple process at the same time. They get preempted according to the use and priority. Ex - Harddisk

### → Virtual Devices

Virtual devices are the combination of dedicated and shared devices. Ex - Printer can be used for printing as well as scanning.

### \* Techniques for Device Management

An operating system manages communication between devices through their drivers. The operating system component provide a uniform interface to access different devices for device management. There are following techniques :-

- Keep track of all devices and programs which is responsible to perform such task these is called I/O controller.

- Maintaining the status of each device such as storage driver, printer and other peripheral devices.
- Allocate and deallocate the device in an efficient way.
- Optimize the performance of devices.

## # Disk Structure

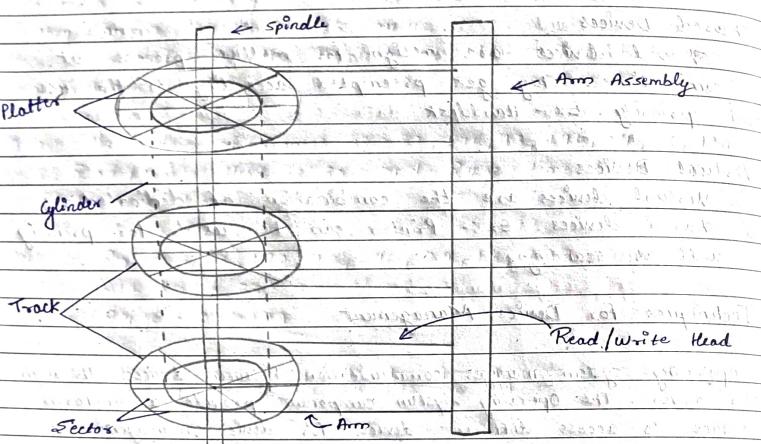


Fig :- Disk Structure

- Data is stored in Hard Disk in form of tracks and sectors.
- Disk surface is divided into tracks.
- Track is further divided into sectors.
- Sector is the addressable unit in disk.

## \* Disk Performance Parameter

There are following component that affect disk performance

### → Disk Access time

The time taken by the disk to complete I/O request is called disk access time.

#### ① Seek Time

The time taken by read/write head to reach the desired track is called the seek time.

#### ② Rotational Latency

The time taken by the desired sector to come under the read/write head is called the rotational latency.

#### ③ Data transfer Rate

The amount of data that passes under the read/write head in a given amount of time is called data transfer rate.

The time taken to transfer the data is called transfer time.

## ④ Disk Scheduling Algorithm

- Disk scheduling is done by OS to schedule I/O requests arriving for the disk.

Disk scheduling is also known as I/O scheduling

→ FCFS (First Come First Serve)

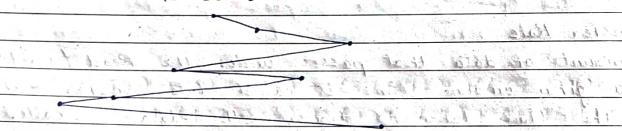
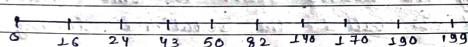
In FCFS, the requests are addressed in the order they arrive in the disk queue.

FCFS is the simplest of all disk scheduling algorithms.

Ques: The order of request is 82, 170, 43, 140, 24, 16, 190  
Read / write Head is 50 and Range time 0 to 199  
find Seek time

Ans:- Request - 82, 170, 43, 140, 24, 16, 190  
R/W Head = 50

Range - 0 to 199

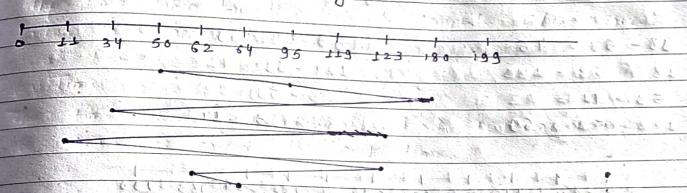


$$\text{Seek time} := (82 - 50) + (170 - 82) + (170 - 43) + (140 - 43) \\ (140 - 24) + (24 - 16) + (190 - 16) \\ 32 + 88 + 127 + 37 + 116 + 8 + 174 = 464 \text{ ms}$$

Ans:- Order of Requests - 95, 180, 34, 119, 11, 123, 62, 64  
R/W Head - 50

Range - 0 to 199 - find the seek time

Ans:- Request - 95, 180, 34, 119, 11, 123, 62, 64  
Read / Head - 50, Range - 0 to 199



Seek time :-

$$(95 - 50) + (180 - 95) + (180 - 34) + (119 - 34) + (112 - 11) \\ + (123 - 62) + (64 - 62)$$

$$45 + 86 + 146 + 85 + 112 + 61 + 2$$

536

### → Shortest Seek Time first (SSTF)

Request having the 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.

Ques Order of Request 98, 183, 41, 122, 14, 124, 65, 67  
Read / write head - 53, Range - 0 to 199 find seek time.

Sol: Request - 98, 183, 41, 122, 14, 124, 65, 67  
R/w head - 53, Range - 0 to 199

$$98 - 53 = 45$$

$$183 - 53 = 130$$

$$53 - 41 = 12$$

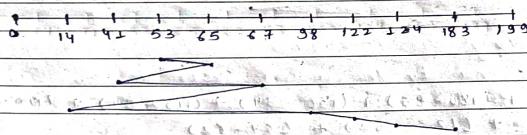
$$122 - 53 = 69$$

$$53 - 14 = 39$$

$$124 - 53 = 71$$

$$65 - 53 = 12$$

$$67 - 53 = 14$$



Seek time :-

$$(65 - 53) + (65 - 41) + (67 - 41) + (67 - 14) + (98 - 14) + (122 - 98) + (124 - 122) + (183 - 124)$$

$$12 + 24 + 26 + 53 + 84 + 24 + 2 + 59 = 284 \text{ Ans}$$

Page No.

Date

Page No.

Date

Ques Order of Request 4, 34, 10, 7, 19, 73, 2, 15, 6, 20  
Read / Write Head - 50, Range - 0 to 100  
find the seek time

Ans Request - 4, 34, 10, 7, 19, 73, 2, 15, 6, 20  
R/w head - 50, Range - 0 to 200

$$50 - 4 = 46$$

$$50 - 34 = 16$$

$$50 - 10 = 40$$

$$50 - 7 = 43$$

$$50 - 19 = 31$$

$$50 - 73 = 27$$

$$50 - 2 = 48$$

$$50 - 15 = 35$$

$$50 - 6 = 44$$

$$50 - 20 = 30$$

$$73 - 50 = 23$$

$$73 - 19 = 54$$

$$73 - 2 = 71$$

$$73 - 15 = 58$$

$$73 - 7 = 66$$

$$73 - 6 = 67$$

$$73 - 4 = 69$$

$$73 - 2 = 71$$

$$73 - 1 = 72$$

$$73 - 0 = 73$$

Seek time :-  
 $(50 - 34) + (73 - 34) + (73 - 20) + (20 - 19) + (19 - 15)$   
 $+ (15 - 10) + (10 - 7) + (7 - 6) + (6 - 4) + (4 - 2)$

$$16 + 39 + 53 + 1 + 4 + 5 + 3 + 1 + 2 + 2$$

$$+ 15 + (16 - 24) + (19 - 18) + (18 - 17) + (17 - 15)$$

$$126 \text{ Ans}$$

### → Scan

In the scan algorithm the disk arm moves in a particular direction and services the request coming in its path and after reaching the end of the disk, it reverse its direction and again services the request arriving its path.

It works as an elevator and hence also known as elevator algorithm.

Ques Order of Request 98, 183, 41, 122, 14, 124, 65, 67  
Read / Write Head - 53, Range - 0 to 199 moving toward larger cylinder.

A). Request - 98, 183, 41, 122, 14, 124, 65, 67

R/W head - 53, Range - 0 to 199

Moving toward larger cylinder

0 14 41 53 65 67 98 122 124 183 199

$$\text{Seek time} : -(53-41) + (41-14) + (14-122) + (122-98) + (98-67) + (67-65) + (65-53)$$

$$+ (53-0) + (0-199) + (199-183) + (183-124) + (124-14) + (14-41)$$

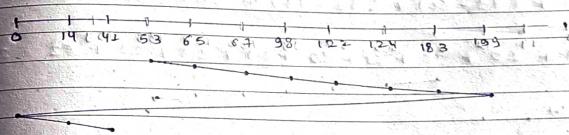
$$12 + 2 + 31 + 24 + 2 + 59 + 16 + 158 + 27 = 331$$

### → C-Scan

In that C-scan algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. The disk arm moves in a circular fashion and this algorithm is also known as Circular scan (C-scan).

Ques Order of Request 98, 183, 41, 122, 14, 124, 65, 67  
Read / write Head - 53, Range - 0 to 199 moving towards larger cylinders.

Soln. Request - 98, 183, 41, 122, 14, 124, 65, 67  
R/W Head - 53, Range - 0 to 199 moving toward larger cylinder



$$\text{Seek time} : -(53-41) + (41-14) + (14-122) + (122-98) + (98-67) + (67-65) + (65-53) + (53-0) + (0-199) + (199-183) + (183-124) + (124-14)$$

$$12 + 2 + 31 + 24 + 2 + 59 + 16 + 158 + 27 = 331$$

331

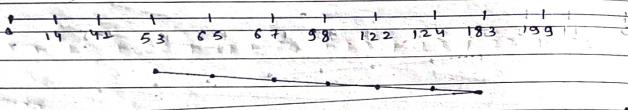
544 Ans

### → Look

Look algorithm is similar to Scan algorithm except the difference is that the disk arm goes only to the last request to be serviced in front of head and then reverse its direction from there only.

Ques Order of Request - 98, 183, 41, 122, 14, 124, 65, 67  
 Read/Write Head - 53, Range 0 to 199 and move head  
 Moving toward largest cylinder

Ans Request - 98, 183, 41, 122, 14, 124, 65, 67  
 R/W head - 53, Range 0 to 199 and move head  
 Moving toward largest cylinder



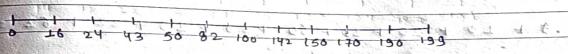
$$\text{Seek time} := (65 - 53) + (67 - 65) + (98 - 67) + (122 - 98) + (124 - 122) + (183 - 124) + (41 - 183) + (12 + 2 + 31 + 24 + 2 + 59 + 142 + 27) = 299 \text{ ms}$$

### → C-Look

C-Look is similar to Scan algorithm. In this algorithm the disk arm is spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end of the last request.

Ques Order of Request - 82, 170, 43, 140, 24, 16, 190  
 Read/Write Head - 50, Range 0 to 199 and move head  
 Moving toward the larger cylinder

Ans Request - 82, 170, 43, 140, 24, 16, 190  
 R/W head - 50, Range 0 to 199 and move head  
 Moving toward larger cylinder



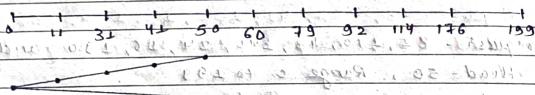
$$\text{Seek time} := (82 - 50) + (100 - 82) + (142 - 100) + (150 - 142) + (170 - 150) + (190 - 170) + (190 - 16) + (24 - 16) + (43 - 24) = 344 \text{ ms}$$

[Date] | [ ]

Que Order of Request - 31, 41, 11, 60, 92, 79, 176, 114  
 Read / Write Head - 50, Range 0 to 199

Moving toward smaller cylinder find the following:-  
 • SCAN • C-SCAN • LOOK • C-LOOK

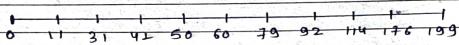
• SCAN



Seek time :-  
 $(50 - 41) + (41 - 31) + (31 - 11) + (11 - 0) + (60 - 0)$   
 $+ (79 - 60) + (92 - 79) + (114 - 92) + (176 - 114)$

$$9 + 10 + 20 + 11 + 60 + 19 + 13 + 22 + 62 = 226 \text{ ms}$$

• C-Scan

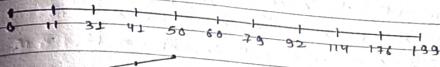


Seek time :-  
 $(50 - 41) + (41 - 31) + (31 - 11) + (11 - 0) + (199 - 0) + (199 - 12)$   
 $+ (176 - 114) + (114 - 92) + (92 - 79) + (79 - 60)$

[Date] | [ ]

$$9 + 10 + 20 + 11 + 199 + 23 + 62 + 22 + 13 + 19 = 388 \text{ ms}$$

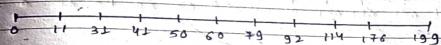
• Look



Seek time :-  
 $(50 - 41) + (41 - 31) + (31 - 11) + (60 - 11) + (79 - 60)$   
 $+ (92 - 79) + (114 - 92) + (176 - 114)$

$$9 + 10 + 20 + 9 + 19 + 13 + 22 + 62 = 204 \text{ ms}$$

• C-Look



Seek time :-  
 $(50 - 41) + (41 - 31) + (31 - 11) + (176 - 111) + (176 - 114)$   
 $+ (114 - 92) + (92 - 79) + (79 - 60)$

$$9 + 10 + 20 + 165 + 62 + 22 + 13 + 19 = 320 \text{ ms}$$

## File-System Implementation

A file system is a method an operating system uses to store, organize and manage files and directories on a storage device.

### → Physical file System

Physical file contains actual data that is stored in computer and describes how data is to be presented or receive from a program. It contains only one record format and one or more members.

### → Logical file System

Logical file system do not contain data it contains description of records that are found in one or more physical files. If it contains more than one format it refers as multifORMAT logical files.

## ④ File System Interface

A file system consists of two parts :-

- A collection of files that store related data.
- A directory structure that organizes and provides information about the files.

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

### \* File Attribute

Files are named to make it easy for users to access them. A file name is a string of characters. The next attribute is extension of that file that is used to identify their format.

### Fundamental Component of file

- ① Name
- ② Identity
- ③ Type
- ④ Location
- ⑤ Size
- ⑥ Protection
- ⑦ Date and Time

### \* File Operation

- Creating a file
- Writing a file
- Reading a file
- Deleting a file
- Truncating a file

V V Step

## # Access Method

When a file is used, information is read and access into computer memory. There are several ways to access these information of files. Some systems provide only one access method.

There are 3 ways to access a file :-

### ↳ Sequential Access Method

In sequential access method data is accessed one record write after another record in an order.

#### Advantages

- It is simple to implement this file.
- Uses lexicography order to enable quick access.

#### Disadvantages

- Become slow if next file record to be retrieved is not present in next record.
- Adding new record may need relocating a no. of records of file.

### ↳ Direct Access Method

Direct access method is also known as relative access method. It allows the program to read and write record rapidly in no particular manner.

The direct access method is based on disk structure method.

Page No.  
Date

Page No.  
Date

#### Advantages :-

- Reducing the average access time of file.
- There is no need to traverse all of the blocks that come before req. block to access the record.

#### Disadvantage

- Difficult to maintain and implement.
- Additional security processes must be put in place.

### ↳ Indexed Sequential Access

It is built on the top of sequential access method. These method construct an index for the file.

#### Advantages

- If the index table is appropriately arranged, it accesses the record very quickly.
- Record can be added at any position in the file.

#### Disadvantages

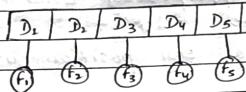
- It is costly and less efficient.
- It needs additional storage space.

## ④ Directory Structure

A directory is a container that is used to contain folder and files it organize file and folder in hierarchical manner.

### ↳ Single Level Directory

Single level directory are used to contain folder in the same directory it has some limitation however when the no. of file increases or the system has more than one user since all the file are in same directory they must have unique name if two user call there file with the same name then unique name rule violated.



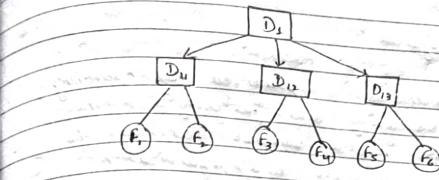
### ↳ Two Level Directory

As we know a single level directory leads to confusion of files name the solution to these problem is to create a separate directory for each user.

In two level directory each user has there own level directory which is known as user file directory (UFD). UFD has similar structure but each list only the files of a single user.

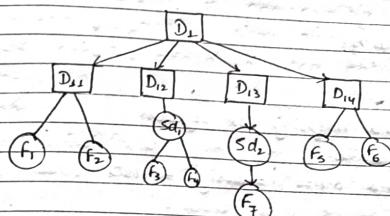
Date \_\_\_\_\_

Date \_\_\_\_\_



### ↳ Tree Structure Directory

This directory structure contain a root directory at the peak the root contain all the directorys for each user. The user can create the subdirectory or files in it a user do not have access to root directory even in this directory user cannot even modify the data.



## ④ Protection In operating System

System protection is a mechanism implemented by operating system to ensure the security and integrity of the system. System protection involve various technique to prevent unauthorized access misuse and modification of operating system and its resource.

↳ Technique to implement protection in Operating system

- User Authentication
- Access control
- Encryption
- Firewall
- Antivirus software
- System update & patches

### \* Advantages

- Ensure security and integrity of system
- Protect sensitive data
- Prevent malware and other security threats

### \* Disadvantages

- Can be complex and difficult to implement
- May slow down system performance.
- Can create additional cost for implementing and maintaining security measures.

## ⑤ Free - Space Management

Free space management involves managing the available storage space on hard disk or other secondary storage device.

The Operating System uses various techniques to manage free space and optimise the use of storage device.

Some of the commonly used free space management techniques are :-

- ↳ Linked Allocation
- ↳ Contiguous Allocation
- ↳ Indexed Allocation
- ↳ File Allocation table (FAT)

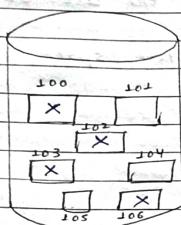
The System keep track of free disk for allocating space to files when it is created. we can reuse this space releasing from deletion of file.

The free space list is maintained by OS which keeps the track of the disk block that are not located to some files or directory.

The free space list can be implemented mainly as :-

#### → Bitvector / Bitmap

Bitmap is a series or a collection of bits where each bit corresponding to a disk block the bit can take two value 0 or 1. 0 indicates the block is allocated and 1 indicates the block is free. It is represented in 16 bit.



Address	Bit value
100	0
101	1
102	0
103	0
104	1
105	1
106	0
106	1

#### → Linked List

In this approach, the free disk block are linked together that is a free block contain a pointer to the next free block.

#### → Grouping

This approach stores the address of the free block in the first free block.

Out of these  $n$  free block the first  $(n-1)$  block are actually free and the last block contain the address of next free block.

#### → Counting

This approach stores the address of first free disk block and the number  $n$  of contiguous disk block that follow the first block. Every entry in these list would contain address of first disk block and a number  $n$ .