I/O devices :

- Hardware, that allows user to interact with the system. They are capable of delivering & receiving data.

Types & characteristics:

* Input devices → used in sending signals to computer.
  eg: keyboard, controller, Joystick, scanner.

* O/P devices → the result of processing from computer is shown via O/P devices.
  eg: displays, printer, speaker, GPS.

OS design issues for I/O management:

- Efficiency → most of I/O devices are slow compared to main memory.
  - ↳ use of multiprogramming lets some processes to be waiting on I/O while other get executed.
  - ↳ Swapping is used to speed up process.

- The quest of generality / uniformality:
  - Ideally handle all I/O devices in the same way.
    - ↳ both in OS & user apps.
  - problem → diversity of I/O ~~applications~~ devices.
    - → different access methods, & vastly diff. data rates.
  - Hides the details of I/O device & we generally see it, read, write, open, close.

Page No.
Date

* Some other issues are.

- Device independence: OS should provide a uniform interface for different I/O devices, allowing applications to interact with them w/o needing to know the specific details of each device.

- Device Drivers: device drivers are crucial software components, that enable OS to communicate with hardware devices.

- Buffering → OS should manage I/O buffers to store data temporarily, that helps increase speed.

- I/O scheduling → using algo's to schedule I/O of the processes.

- Error Handling → Robust error handling mechanisms are required to detect & recover from errors.

- Direct Memory Access (DMA) → Allows device to transfer data directly to & from memory w/o CPU intervention, reducing CPU overhead & improving I/O speed

- I/O control blocks → These Data structure contains info about I/O requests like device status, buffers, etc.

- Synchronisation → coordination b/n process sharing I/O devices is necessary to avoid conflicts & ensure data consistency.

Page No.
Date

- Interrupt handling → mechanisms to handle these quickly.
- Virtual file system → a hierarchical file system to manage files & directories is imp.
- Caching → using cache memory to store frequently accessed data can improve I/O performance.

* I/O Buffering :

- A buffer is a memory area that stores data being transfered betn device and application.

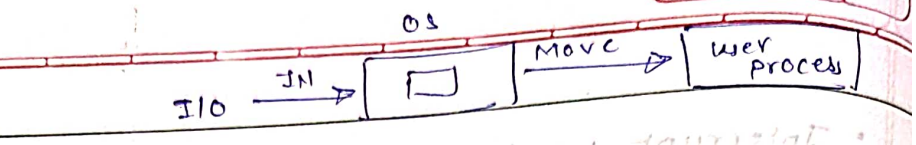
* uses of I/O buffer:

- Buffering is done to deal effectively with a speed mismatch betn producer & consumer of data stream.
- A buffer is produced in main memory to heap up the bytes received from modem.
- Data gets transfered to disk from buffer.
- The process of data transfer is not instantaneous, thus modem requires another buffer to handle incoming data.
- When both buffers complete their task, modem switches back to 1st buffer, while data from 2nd gets transfered to disk.

* Types of Buffers:

a) Single Buffer ⇒

Buffers is provided by the OS, to the system portion of main memory.



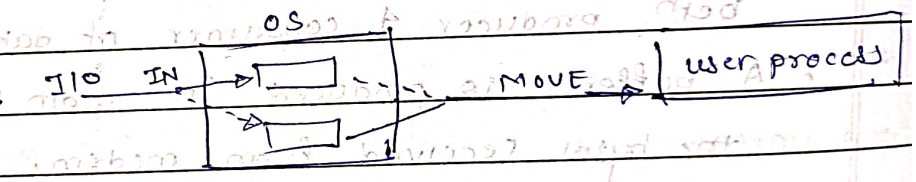
Block oriented :

- system buffer takes i/p
- after i/p, block gets transfered to user space & the process requests for another blocks
- 2 blocks work simultaneously
- OS can swap processes.
- OS can record the data of system buffer to user process.

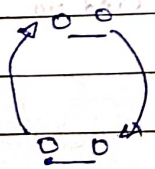
Stream oriented :

- line at a time operation is used for scroll mode terminals. User inputs one line at a time with a carriage return signaling at end of line.
- Byte - at a time operation is used on form mode, terminal when each keystroke is significant.

(b) Double buffer :



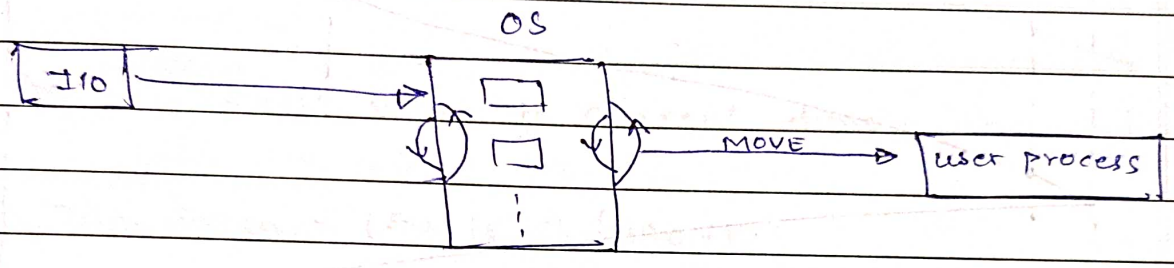
- Block Oriented :
  - 2 buffers available
  - one used by driver / controller to store the data while waiting for it to be taken to higher level hierarchy.
  - other is used to store data from lower level module.
  - Also known as buffer swapping.



- Stream oriented
- Disc - at a time I/O, user process needs to be suspended for i/p or o/p, unless process runs ahead of double buffer.
- Byte - at a time operations, double buffer offers no advantage over single buffer.

(c) Circular Buffer ⇒

- When more than 2 buffers are used, that collection of buffers is called circular buffer.
- Here, data is not directly passed from Producer to consumer, as data would change due to overwriting of buffers.
- The producer can fill up only 1 buffer i.e. i-1, while data in i is waiting to be consumed.



Disk scheduling algorithms:

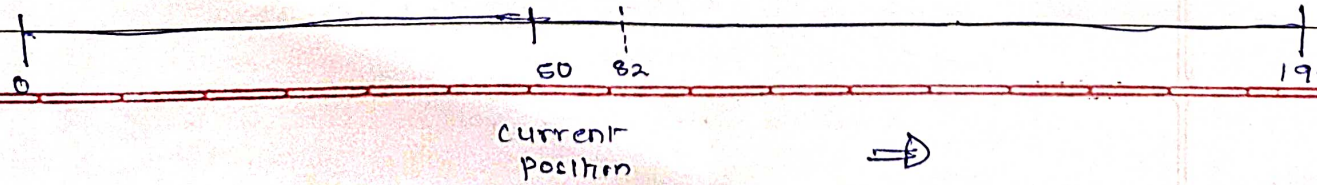
- To minimize seek time i.e. time taken to reach desired track
- Platter = CD's, that has upper & lower surface, these surface has lines = tracks, & each track creats sectors.

* . Platter → Surface → track → Sector

* FCFS

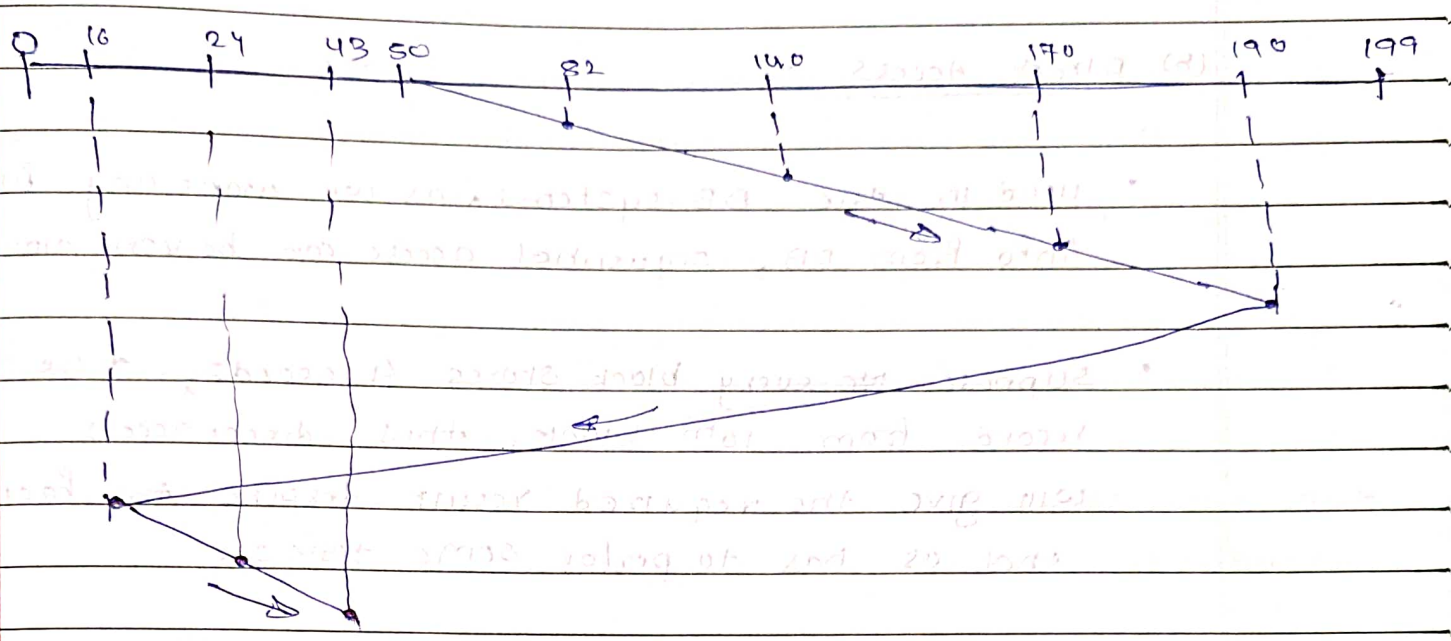
Q) disk has 200 tracks (0-199). Request queue contains tracks (82, 170, 43, 140, 24, 16, 190)
Current position of head = 50.

- Calc. ~~total~~ total no. of tracks movement by R/W head?



100°C

we are given direction to start → move towards largest value not extreme then come back to smallest value & do not accept anything. & then again go & accept values remaing.



→ (190 - 50) + (190 - 16) + (43 - 16) = 341.

File Management

* A file can be defined as a data structure, that stores sequence of records, stored on the system.
- The collection of files is known as directory & collection of directory at different level is known as File system.

File Access Methods:

* We have got 3 methods to do this:

a) Sequential access: (word by word)

- Most of the os, access the file sequentially.
- In sequential access, os reads the file word by word. A pointer is maintained which initially points to base address of the file. If the user wants to read the 1st word of file then ptr provides that word to the user and increases its value by 1 word. This process continues till end of file.

- Directory entry contains a ptr to starting & ending of file block & each block has ptr to next block.

Adv → very flexible in terms of size & no compulsion of contagious memory.
→ does not suffer from external fragmentation.

disadv → a large no. of seeks are required to access entire data
→ no sappot of random or direct access. only sequential allocaed.
→ pointers are used.

(c) Indexed allocation:
- A special block is included (Index Block) contains the pointer to all blocks occupied by a file.
  Each file has its own index block.
- The entry in index block contains disk address of the file block.

Adv → we get direct access to required block.
→ we overcome external fragmentation.

disadv → pointer overhead for indexed allocation is greater.
→ for some cases inefficient.

Page No.
Date

Security:

- The security to Files can be provided by managing the access to the files.

The files that have direct access of any user need protection. Access can be given to users accordingly
→ Read, write, Execute are types of permissions.
- Append → write more info in file
- Delete, list → list name of attributes

other operations of a file will / can also be controlled.

* Access control →

- There are many methods via which many users would access the file, thus list all files & directories is called access control lists (ACL), specifying the name of users and types of access associate with each of user.

- To condense the size of ACL, we use 3 classifications:
→ owner ⇒ creator of file.
→ Group ⇒ grp. of people who have similar action with file.
→ universe ⇒ all users undergo category called univers.

* other Protection approaches:

The access to any system is controlled by passwords, they may result to effective access to file.
But passwords are hard to manage & store.

(c) JFS ⇒ Journaled file system, alternative to EXT file system & can be used in place of EXT 4

- As stability is needed with few resources.
- A handy file when CPU power is limited.

(d) XFS → considered as highspeed JFS; that is developed for parallel I/O processing

- Nasa still uses this with its high storeage servers

(c) ReFS → Resilient file system, designed to overcome NTFS.

- ReFS uses a B+ tree structure to manage data through metadata indexing.
- A B+ tree stores data in a branching pattern, where each node in the tree has an ordered list of keys & pointers to lower level nodes or leaves.
- ReFS has several key feature, that make it a reliable & efficient file system.
- One of the feature is to maintain data integrity & other feature is to focus on data availability.

5.5 Linux file system → A built in layer in linux os. used to handle data management of storeage.

a) EXT, EXT2, EXT3 & EXT4 : Extended file system.

- primarily developed for MINIX OS. This is an older version & generally not used today.
- EXT2 is the 1st linux file system that allows managing 2 T.B. of data
- EXT3 an updated version of EXT2 and has backward compatibility but does not sappot servers.
- EXT4 is fastest amongst all & an considerable opt for SSD.

b) Reiser File system: Reiser is an alternative to EXT3 file system.

- Being improved performande & advance ~~result~~ features.
- The file system dynamically sapports the file extension, but has some drawbacks in performances.

blocks may hold

cache

Page No.
Date

5.4

Windows File System

We get (a) file systems in windows,

(a) FAT → File Allocation table developed by Microsoft in 1977.
FAT family includes FAT, 12, 16, 32 & exFAT.
• FAT is still a common format for use sticks and external hard drives

• FAT 12 is used in floppy disks & FAT 16 is mostly used in mobile with max size of 4 Gigabytes.

• FAT 32 is also suitable for all mobile phones.

• The abbrevation exFAT stands for Extended File Allocation Table and was produced in 2006 especially for flash memory.
Max file size is 512 Terabytes.

(b) NTFS → New Technology file system & has versions 1.0, 1.1, 2, 3.0, 3.1
• Internal hard drives can be formatted since windows vista only with NTFS.
• The max file size of NTFS file system is theoritically 16TB.
• NTFS is more secure than FAT, user & user-group can be given permission to read, write & execute.

...ndows ... blocks.

Page No.
Date

Free Space Management :

A File system is responsible to allocate free blocks to the file, thus it needs to keep a track of all free blocks

Two methods

a) Bit Vector →

- Here, free space is implemented as a bit map vector, it contains no. of bits where each bit represents a block.
- If block is empty bit is 1 else 0.

According to this the blocks are then allocated.

b) Linked list →

- This suggests linking all free blocks together & keeping a pointer in cache that points to 1st free block.
- Thus all blocks are linked & will get allocated.

this is an issue, variable ... identical memory blocks.

* We generally need to use trusted sources for file sharing to reduce the risk of malware & viruses.
* To reduce data leaks & breaches, the file access should be controlled.

File Allocation

Allocation decides how files are stored in disk blocks. One main reason behind this is efficient disk space utilization & fast access.

3 methods

(a) contagious allocation:

- Each file occupies a contagious set of blocks on disks. Thus, being contagious, we can determine blocks occupied by file.
- The directory entry for a file with contagious allocation contains:
  - → Address starting block
  - → length of allocated portion.

Adv → both sequential & direct access is allowed.
→ extremely fast

disadv → suffers from internal & external fragmentation.
→ file size is a problem, coz of contagious memory allocation.

(b) linked list allocation:

- Here each file is in linked list, which need not be contagious, the disk blocks can be scattered anywhere on the disk.

Page No.
Date

- But file sharing comes with potential risks like spread of malware & viruses, data leaks, etc.

Types of file sharing:
(Practice of providing digital access to digital files).

1) Peer to Peer (P2P) → allows to share files with each other w/o need of centralized server. Instead users connect to each other directly & exchange files over network of peers.

2) Cloud based → Make file stored on a external remote server, that can be accesed by other devices via internet. Easy method to share & access files.

3) Direct file transfer → transfer of files betn devices that are directly connect via some medium (bluetooth).

4) Removeable media file sharing → use of physical storeage devices (USB) to share data amongst users.

* Risks in file sharing:

- Malware & viruses → most common issue in P2P & any docoloads via unwanted / unknown sources.
- Data breach & leak → can be observed in cloud based & P2P
- legal consequences → non copyright, may lead to this. & lastly Identity theft.

* Tree structured directories →

- Here any directory can be a file or a directory/ sub-directory.
  The similar type of files can be grouped.
- Each user has its own directory & cannot enter into other users & can have watch the roots directories, but cannot modify them.
- Searching is quite efficient here.
- It may have various levels of permissions; that are R W X, Read write execute.

* Acyclic Graph structure directories:

- Tree wont allow same file to exist in multiple dirs. Thus we use acyclic graph.
- Here, two or more directory can point to same file

File Sharing :

- File sharing is the practice of distributing or providing access to digital files betn two or more users or devices.
- File sharing plays a vital role in facilitating collaboration and communication among individuals and organisations.
  Allowing people to share files quickly & reducing remote work.

* File management comes with some objectives like,
  - File organisation, Data security, sharing, backup, File compression, File encryption, file removal, Spacemanagement, etc.

File Directories

- Directory can be defined as listing of related files on disks. The directory may store some or the entire file attributes
- A hard disk can be divided into various sections, it can be called as mini disks.
- Each partition must have atleast one directory in which, all files of partitions can be listed.

* Directory can be called as a file that has meta data of other files.
- We can get file creation, deletion, transfer, search is possible.

* Single level Directory ⇒ A simple method to have one big list of all files on the disk
  - Entire system will have only 1 Directory, & all files are included in it.

* Two level Directory → we can create a seperate directory for each user.
  - We have one main master directory which contains seperate directories ~~def~~ dedicated to each user
  - System wont let one user to enter to other users file.
  * diff. username thus can have same file name.

ntify this by same

iting operation seems instantly done.

and value change is seen in all processors in same order.

coherent behaviours.

(b) Direct Access →

- used in the DB systems, as we want any filtered info from DB, sequential access can be very slow.
- suppose every block stores 4 records, & we need record from 10th block, thus direct access will give the required result despite the fact that OS has to perfor some tasks.

(c). Indexed Access →

- If a file can be sorted on any of the field, then an index can be assigned to a grp of certain records.
  Thus a particular record can be accessed by index.
- Index is nothing but address of record in a file.

File management overview:

- set of processes and techniques involved, in creating organising, accessing, manipulating & controlng files stored on storeage devices such as hard drives, SSD's or network storeage.
- It acts as an intermediary layer betn applicahons & underlying storeage hardware, providing a logical & organised structure for storing & retrng data.