Operating System

What is an operating system?

A program that acts as an intermediary between a user of a computer and the computer hardware.

Operating system goals:

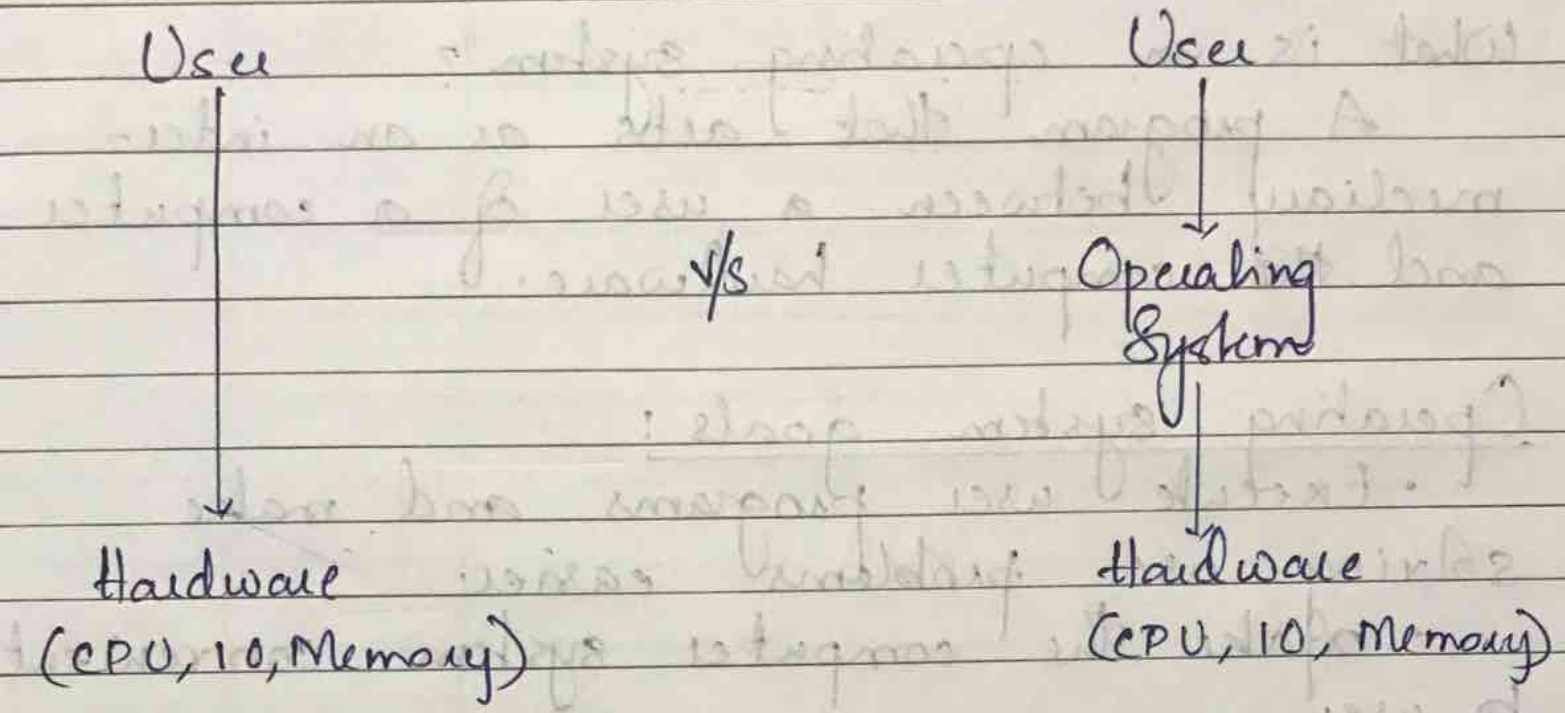
- Execute user programs and make solving user problems easier.
- Make the computer system convenient to use.
- Use the computer hardware in an efficient manner.

Computer System Structure

> Hardware > Operating System > Application programs

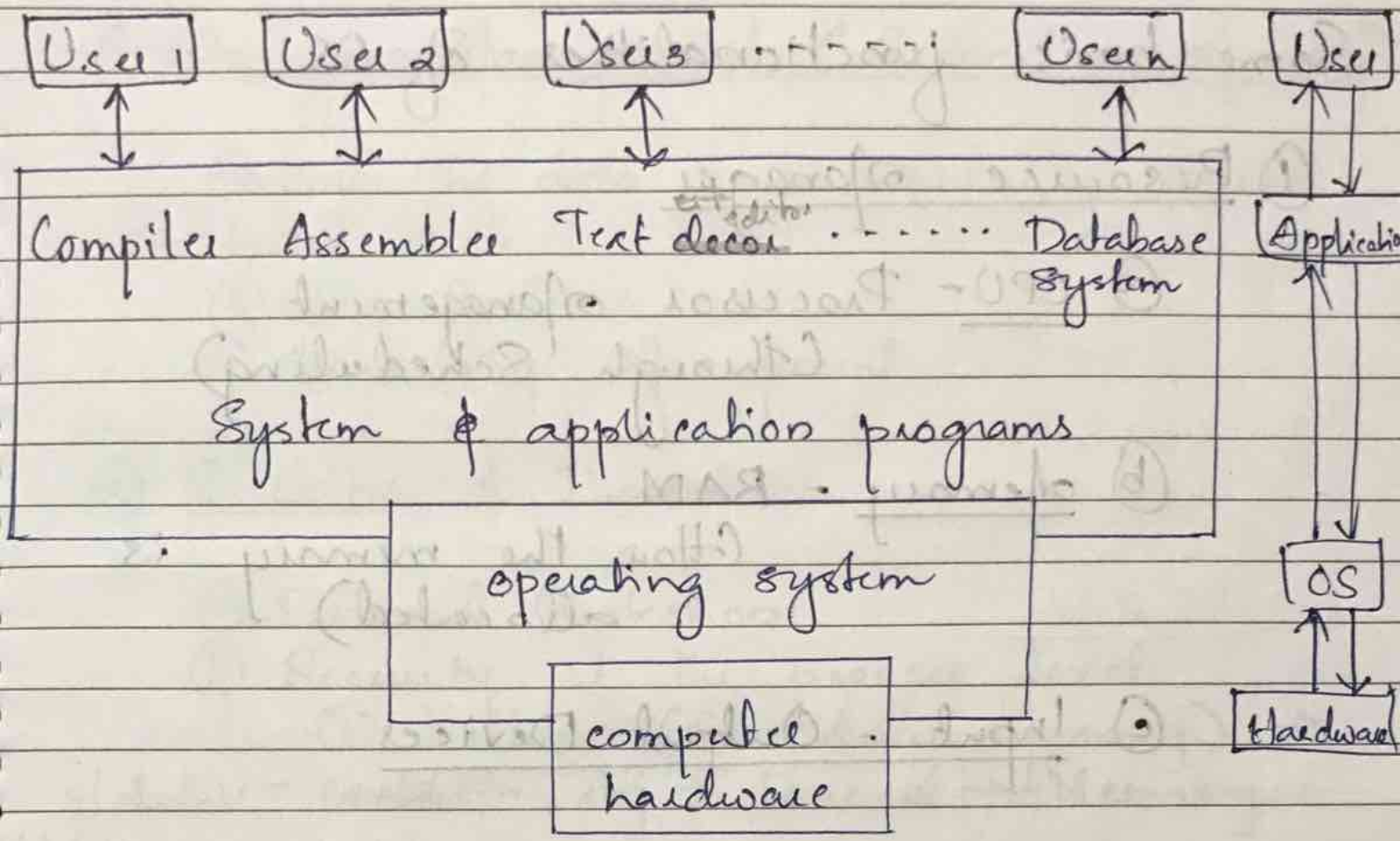
Users

Why we need Operating System?



Goal Of Operating System

1. Convenience
   - a. Easy to Use (Users friendliness) 85% windows
2. Efficiency - Linux software
   - a. Throughput (No. of task performed per unit time)



The dominant desktop OS is Microsoft Windows with a market share of around 82.74%. macOS by Apple Inc. is in second place (13.23%), and the varieties of Linux are collectively in third place (1.57%).

Some basic functionalities of OS

(1) Resource Manager

(a) CPU - Processor management (through Scheduling)

(b) Memory - RAM (How the memory is allocated)

(c) Input - Output Devices

RAM - Volatile
ROM - Non Volatile

(2) Memory Management

(a) Space for the process should be created in RAM

(b) Example of hello.c program
(i) Hello.c (Secondary memory) → hello.obj → Process → Memory

(3) I/O Devices Management

① Storage Management (Secondary Memory)

ⓐ How the data is permanently stored in hard disk.

ⓑ File Systems

ⓒ Directory Management

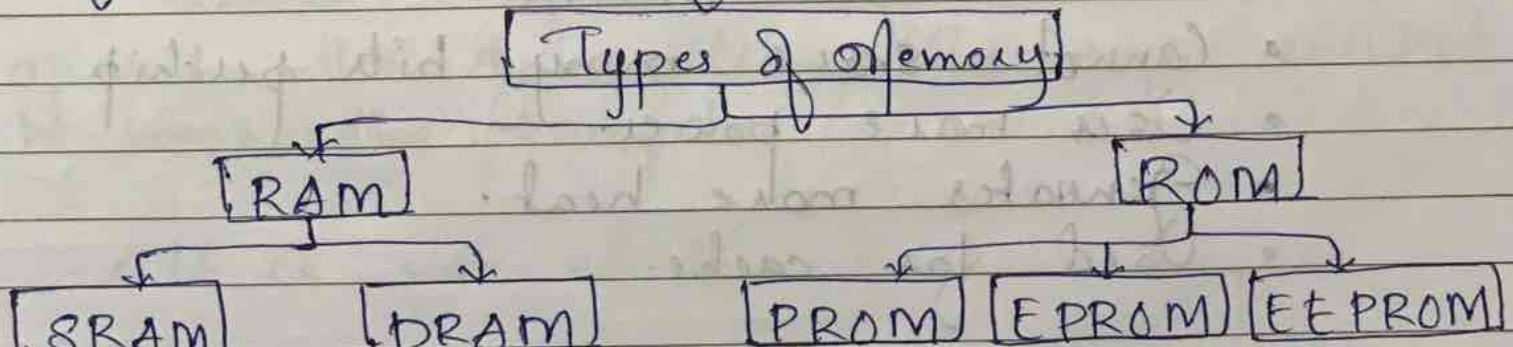
② Security & Protection

ⓐ Password Protection

ⓑ Security at the process level

(i) Hello.c (Secondary Memory) → Hello.obj → Process → Memory

Different types of RAM and ROM ?



(i) RAM (Random Access memory)

- main memory or the primary memory
- The program that requires during the execution of a program are stored in this memory.

> SRAM

- Constructed of circuits similar to D flip flops.
- Holds its contents as long as power is sustainable.
- Expensive
- Faster than DRAM
- Cannot store many bits per chip
- Uses more power.
- Generates more heat.
- Used for cache.

> DRAM

- Constructed of tiny capacitors that leak electricity.
- Requires a recharge every few milliseconds to maintain its data.
- Inexpensive
- Slower than SRAM
- Can store many bits per chip
- Uses less power
- Generates less heat.
- Used for main memory.

(ii) ROM (Read Only Memory)

- Stores crucial info essential to operate the system, like the pgm essential to boot the computer
- It is not volatile
- Always retains its data
- Used in embedded systems or where

the programming needs no change.

- Used in calculation and peripheral devices.

- ROM is further classified into 4 types - MROM, PROM, EPROM, & EEPROM

> PROM (Programmable read Only memory)

It can be programmed by the user. Once programmed, the data & instruction it cannot be changed.

> EPROM (Erasable Programmable read-only memory)

It can be programmed. To erase data from it, expose it to ultraviolet light to reprogram it, erase all the previous data.

> EEPROM (Electrically Erasable Programmable Read Only Memory)

The data can be erased by applying an electric field, with no need for ultraviolet light, we can erase only portions of the chip

> MROM (Marked ROM)

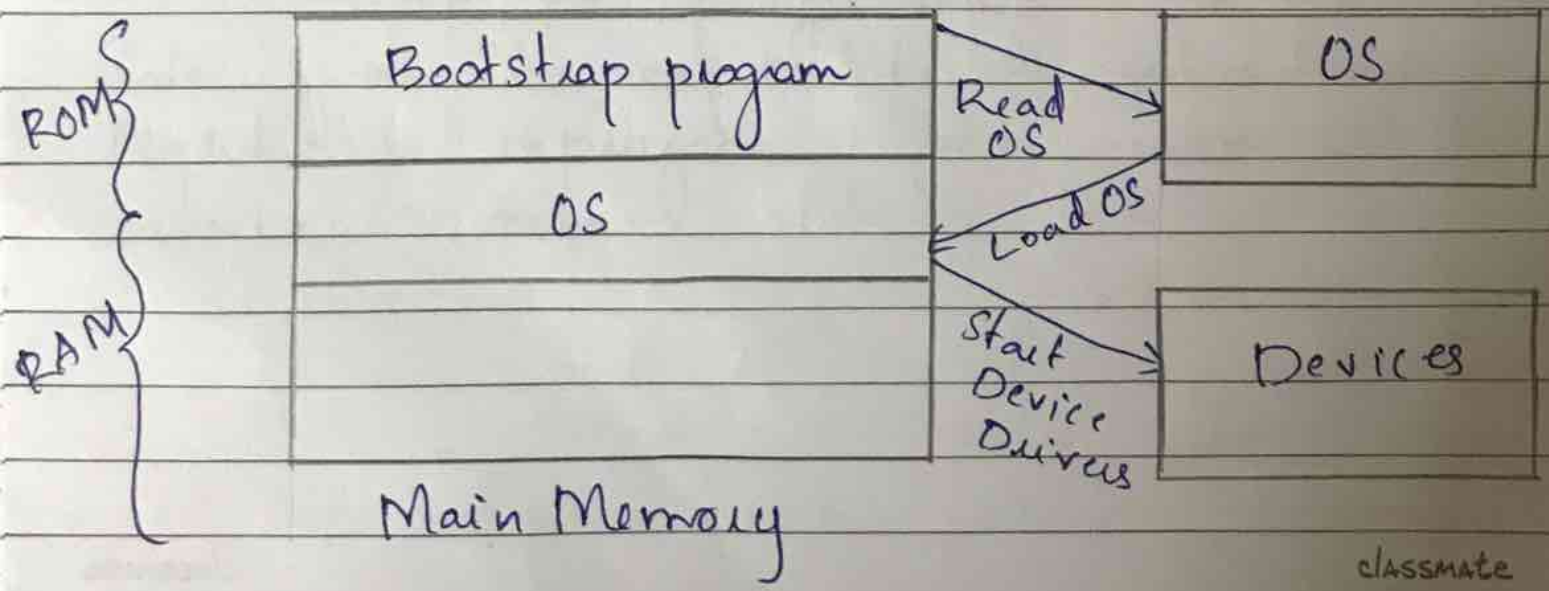
The very first ROM were hard-wired devices that contained a pre-programmed set of data or instruction. These kind of ROMs are known as marked ROMs which are expensive.

Agenda
- Bootstrap program
- Storage Types
- Types of OS
- Introduction to Files

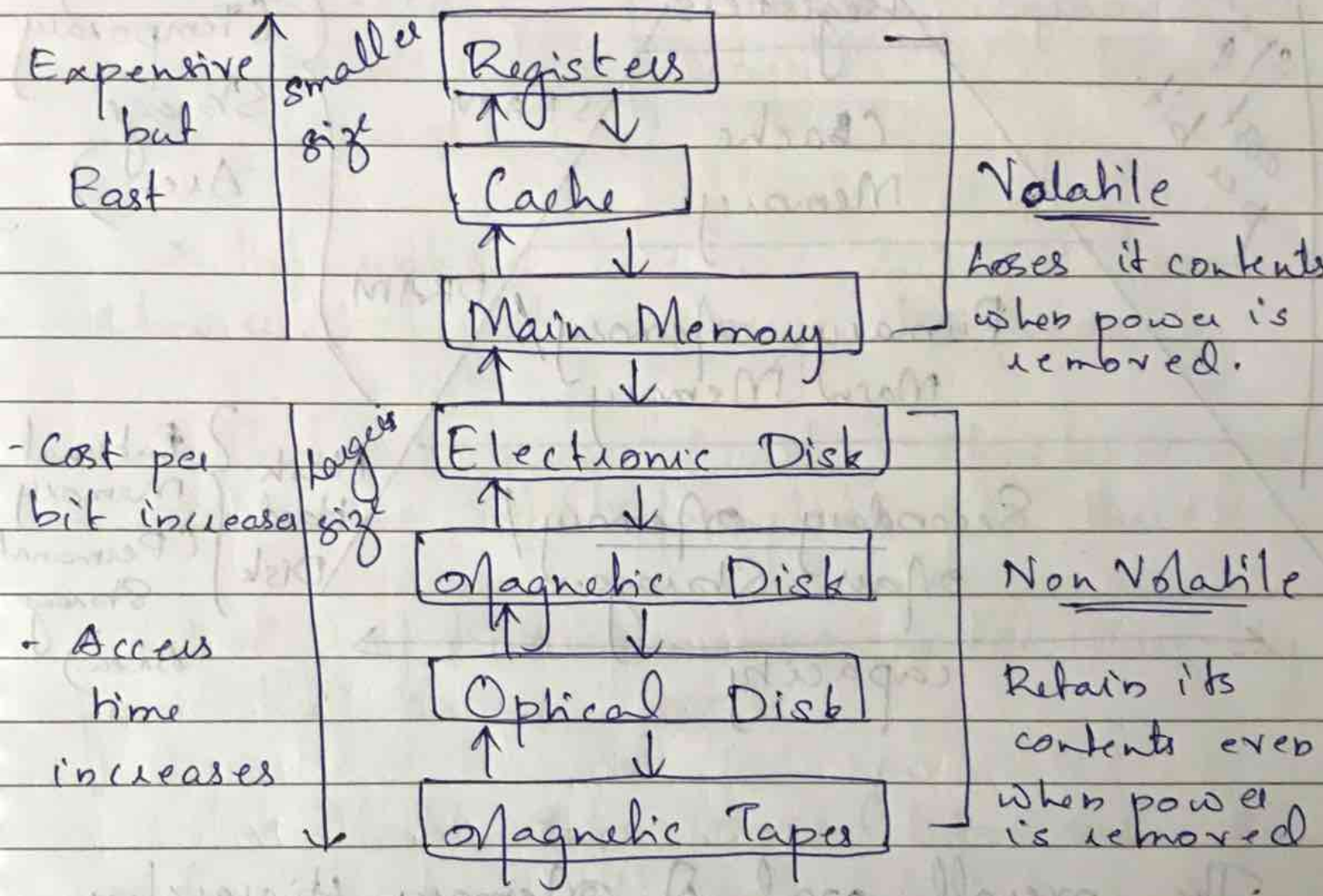
Computer Startup

> Bootstrap program is loaded at power-up or reboot

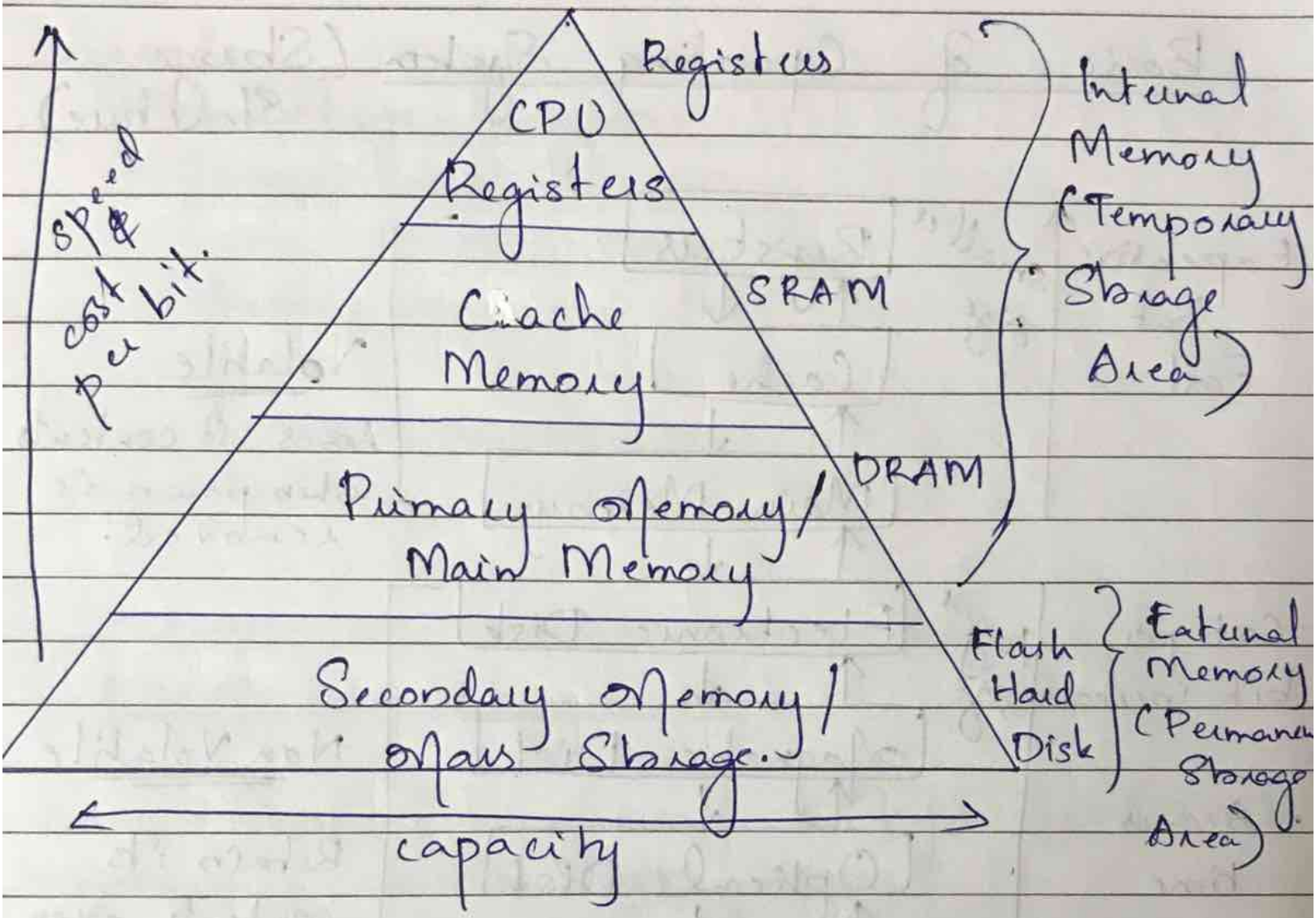
> Typically stored in ROM or EEPROM, generally known as firmware
> Initialize all aspects of system
> Loads operating system kernel & starts execution.



Basic of Operating System (Storage Structure)



classmate



The overall goal of memory hierarchy is to obtain the highest possible average access speed while minimizing the total cost of the entire memory system.

Caching

> Important principle, performed at many levels in a computer (in hardware, OS, Software)

> Information in use copied from slower to faster storage temporarily

> Faster storage (cache) checked first to determine if information is there

- If it is, information used directly from the cache (fast)
- If not, data copied to cache & used there.

Different Types of OS

- Batch OS
- Multiprogramming OS
- Multi-Tasking OS
- Multi-Processing OS
- Real-time OS

(1) Batch Operating Systems

- What do you mean by a batch?
  - Example of Student Batches.
  - In the case of OS, Batch = Batch = multiple jobs
  - Jobs = Program + Input Data + Control Instructions

- **Batch Operating Systems are non-interactive.**
  - User cannot directly interact with the system.

- User will prepare jobs in punch card and loaded into the computer with the help of an operator.

CPU utilization is very low so loading & unloading takes time.

② Multiprogramming (Round Robin) Operating Systems

- A computer running more than one program at a time (like running Excel & Firefox simultaneously).

- In a modern computing system, there are usually several concurrent application processes which want to execute.

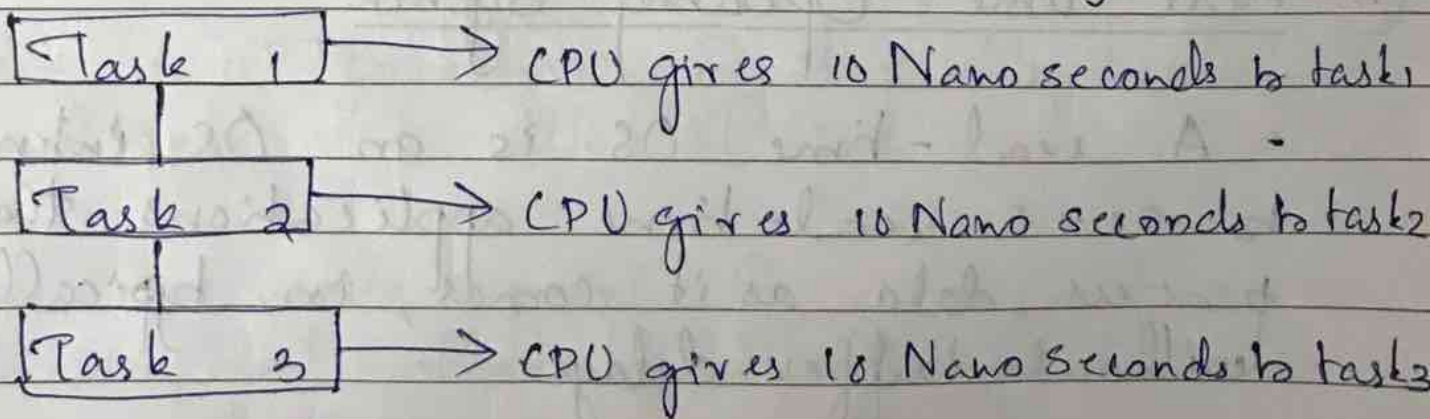
- Now it is the responsibility of the OS to manage all the processes effectively & efficiently.

- The main memory is too small to accommodate all of these processes or jobs into it.

- These processes are initially kept in an area called job pool.
- One CPU and multiple processes
- CPU time will be allocated to a single process at a time.
- At some stages, if the admitted process needs some resources, for example, I/O devices.
- Then that process will be temporarily suspended from the CPU & the CPU time will be allocated to the next process in the ready queue.
- Please note that the user cannot suspend a running process and only CPU can do that.

③ Multitasking Operating System

- Multitasking OS is an extension of multiprogramming OS
- Multitasking OS is called multiprogramming with Round Robin.
- Each process will get a time quantum
- The process has to complete its execution within the time quantum.
- If the process cannot complete its execution within the time allocated, then that process has to move out of the CPU and enter back into the ready queue.



④ Multiprocessing Operating System

- Multiple CPUs and multiple processes
- Each process will be allocated to the CPU
- The main advantage of multiprocessor system is to get more work done in a shorter period of time.
- These types of systems are used when very high speed is required to process large volume of data.

⑤ Real Time Operating System

- A real-time OS is an OS intended to serve real-time applications that process data as it comes in, typically without buffer delays.

11 10 7

- The response time is very less as compared to other OS.

- There are 2 types of Real Time OS:
  - Hard Real Time OS — Process has to complete in exact time, eg: Time Bomb
  - Soft Real Time OS — within a particular time

File System in OS

What are files?

- A file is a collection of related information (sequence of bits) that is recorded on secondary storage.

- File is a collection of logically related entities.

- From a user's point of view, a file is the smallest allotment of logical secondary storage.

Types:

> Data
  - numeric
  - character
  - binary
> Program

File Attributes - Properties

> Name :- only information kept in human-readable form.

> Identifier :- unique tag (number) identifies file within file system

> Type :- needed for systems that support different types. .doc .pptx

> Location :- pointer to file location on device.

> Size :- current file size.

classmate

> Protection :- controls who can do reading, writing, executing. - permission

> Time, date, and user identification :- data for protection, security & usage monitoring

File Management

The main objectives of the file management system are:

- It provides I/O support for a variety of storage device types.

- Minimizes the chances of lost or destroyed data.

- It provides I/O support for multiple users in a multiuser systems environment.

Properties of a File System

The properties of a File System are:

- Files are stored on disk or other storage and do not disappear when a user logs off.

- Files have names and are associated with access permission that permits controlled sharing.

- Files could be arranged or more complex structures to reflect the relationship between them.

File Operations

> File is an abstract data type

- Create
- Write
- Read
- Delete Entire file with the attributes

- Truncate — The attributes will remain but the contents of the file will be deleted.

File Types - Name, Extension

File Type	Usual extension	Function

Access Methods

(1) Sequential Access — Magnetic Tapes Based

one by one

read next
write next
reset

(2) Direct Access — Disk Based
in different blocks

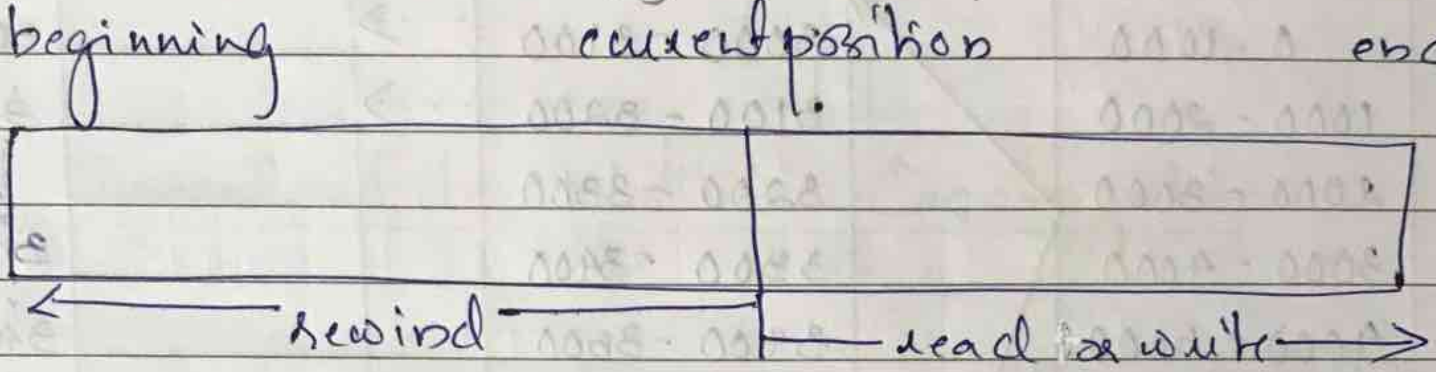
read n
write n
position to n
read next
write next
rewrite n

n = relative block number

(3) Indexed Sequential Access — Disk Based

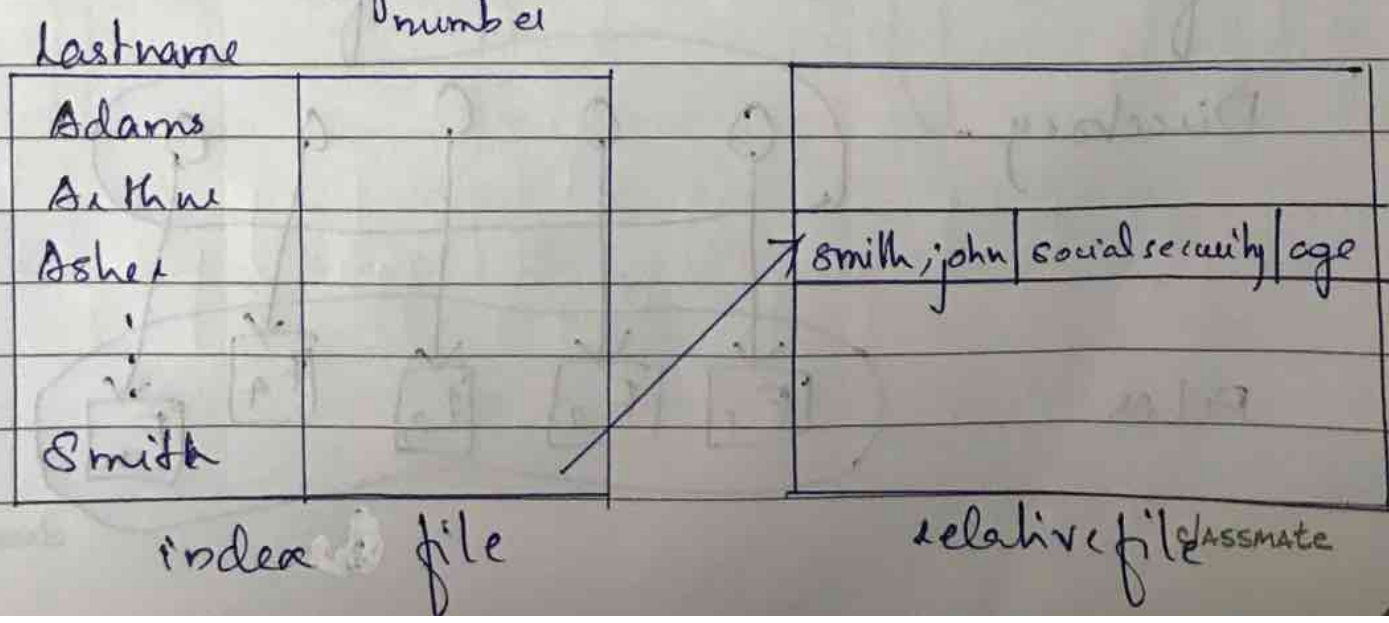
Sequential - access File

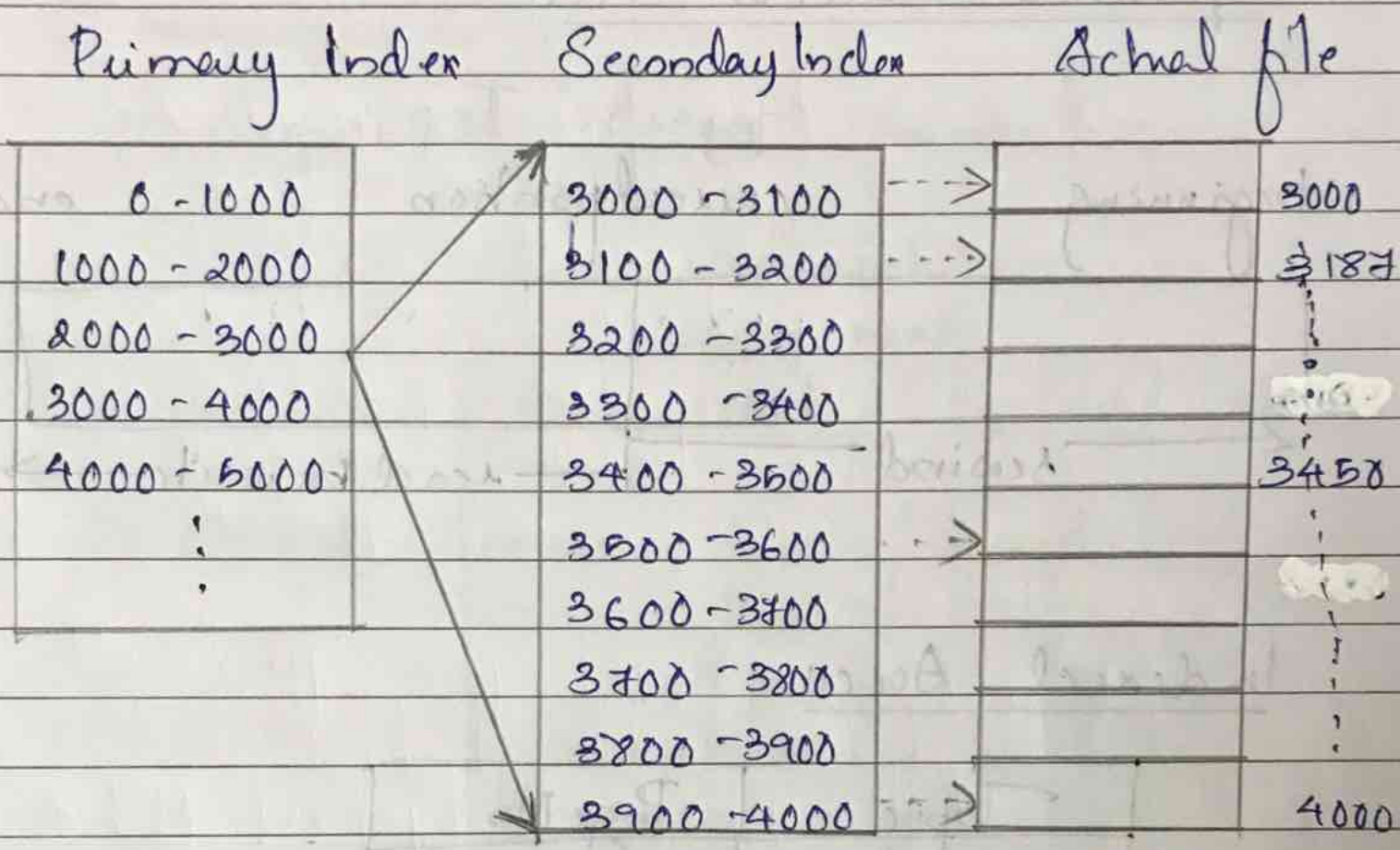
Magnetic Tapes



Indexed Access

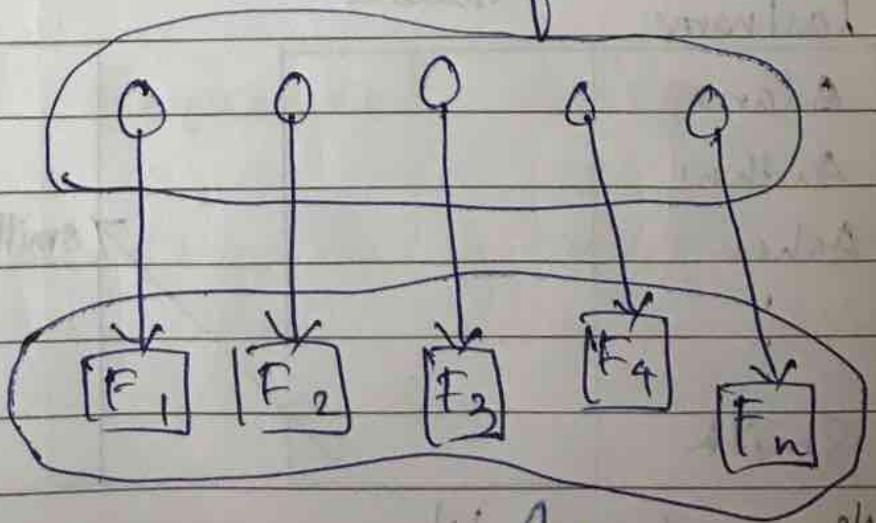
Topic	Page No
C	276
C++	238
Array	402





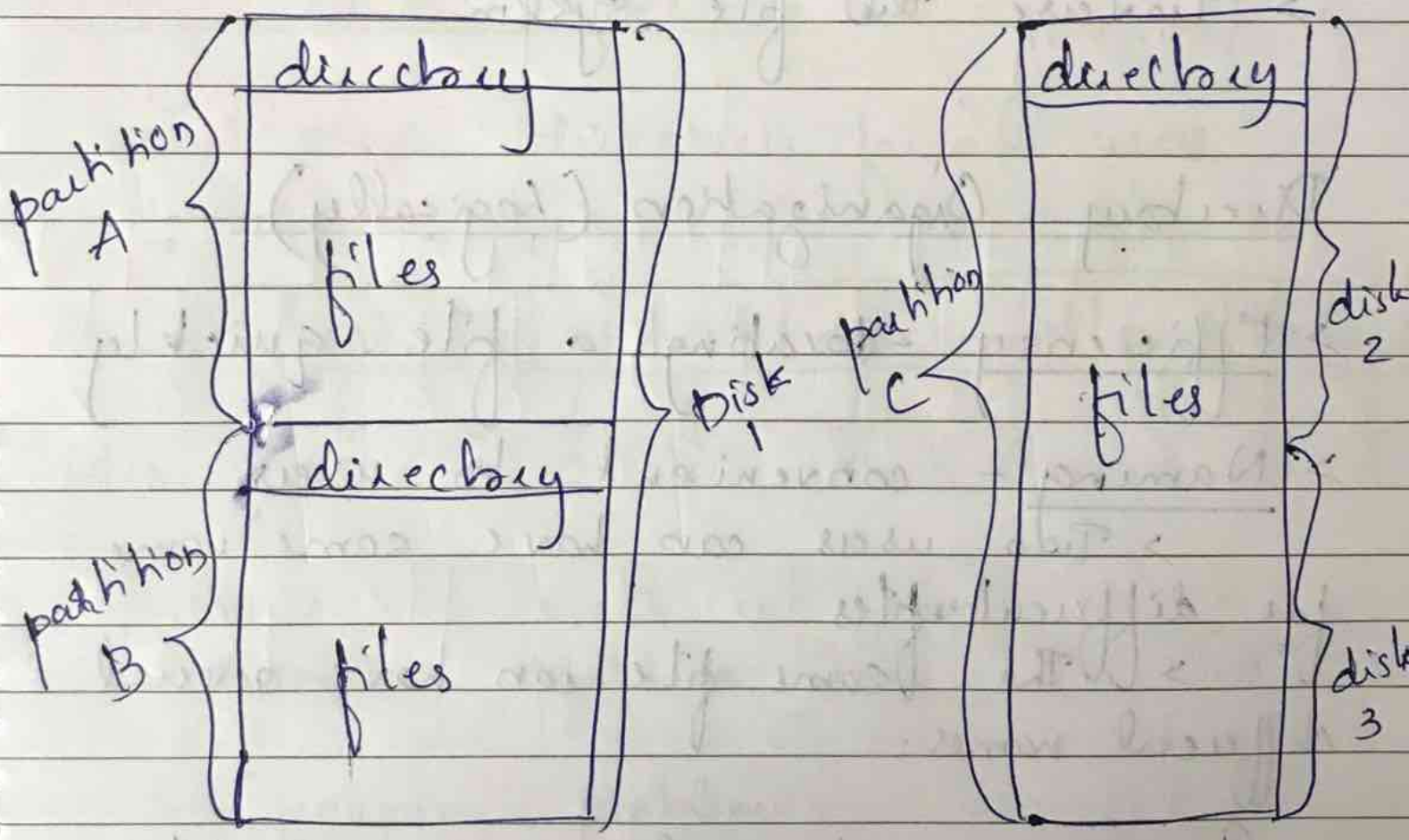
Directory Structure

> A collection of nodes containing information about all files



Both the directory structure and the files resides on disk Backups of these two structures are kept on tapes.

A Typical File-system Organization :



Operations Performed on Directory

> Search for a file.
> Create a file
> Delete a file
> List a directory
> Rename a file
> Traverse the file system

Directory Organization (logically)

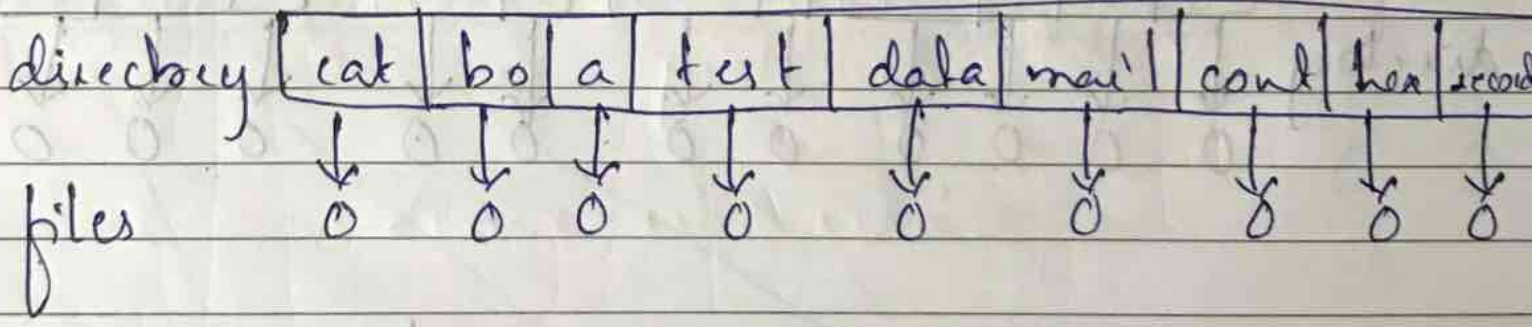
> Efficiency - locating a file quickly
> Naming - convenient to users.
  > Two users can have same name for different files
  > The same file can have several different names.
> Grouping - logical grouping of files by properties, (eg, all java programs, all games, ...)

Types Of Directories

1. Single Level
2. Two Level
3. Tree
4. Acyclic

Single Level Directory

> A single directory for all users



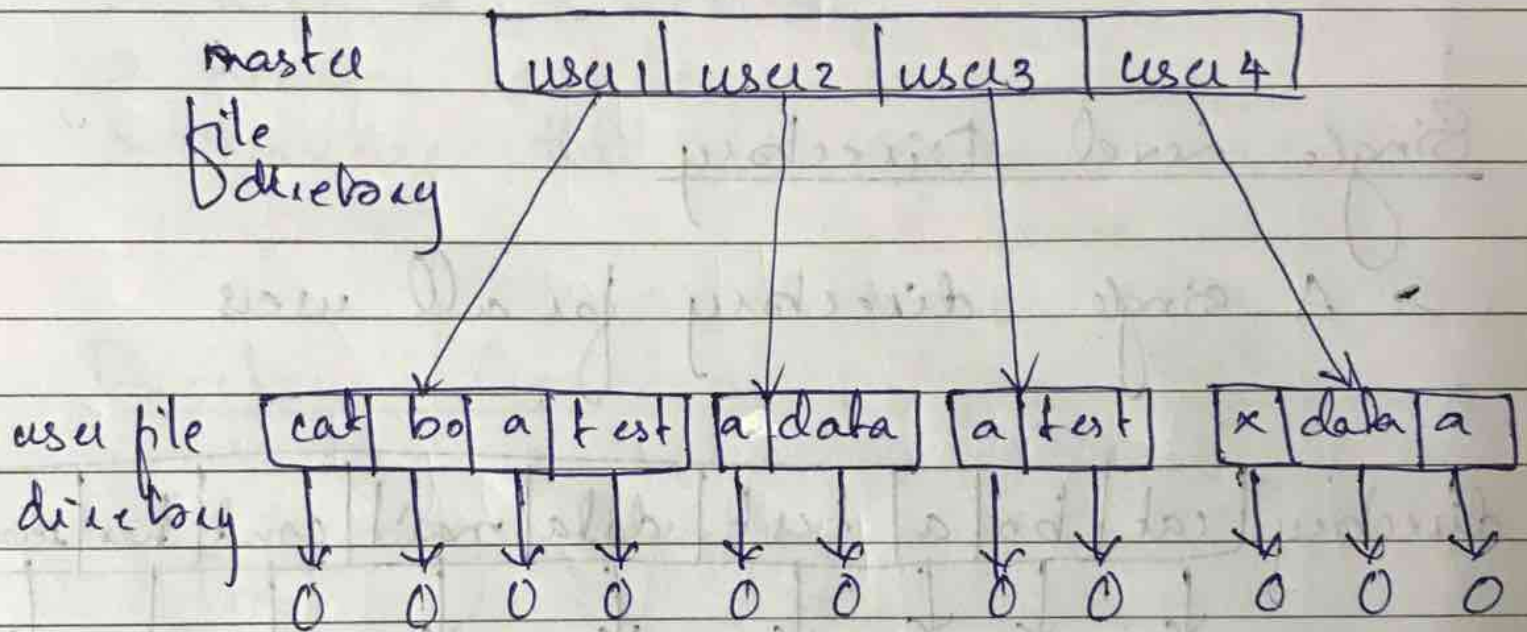
> Naming problem

> Grouping problem

No two users can have a single name in single level directory.

Two - Level Directory

> Separate directory for each user

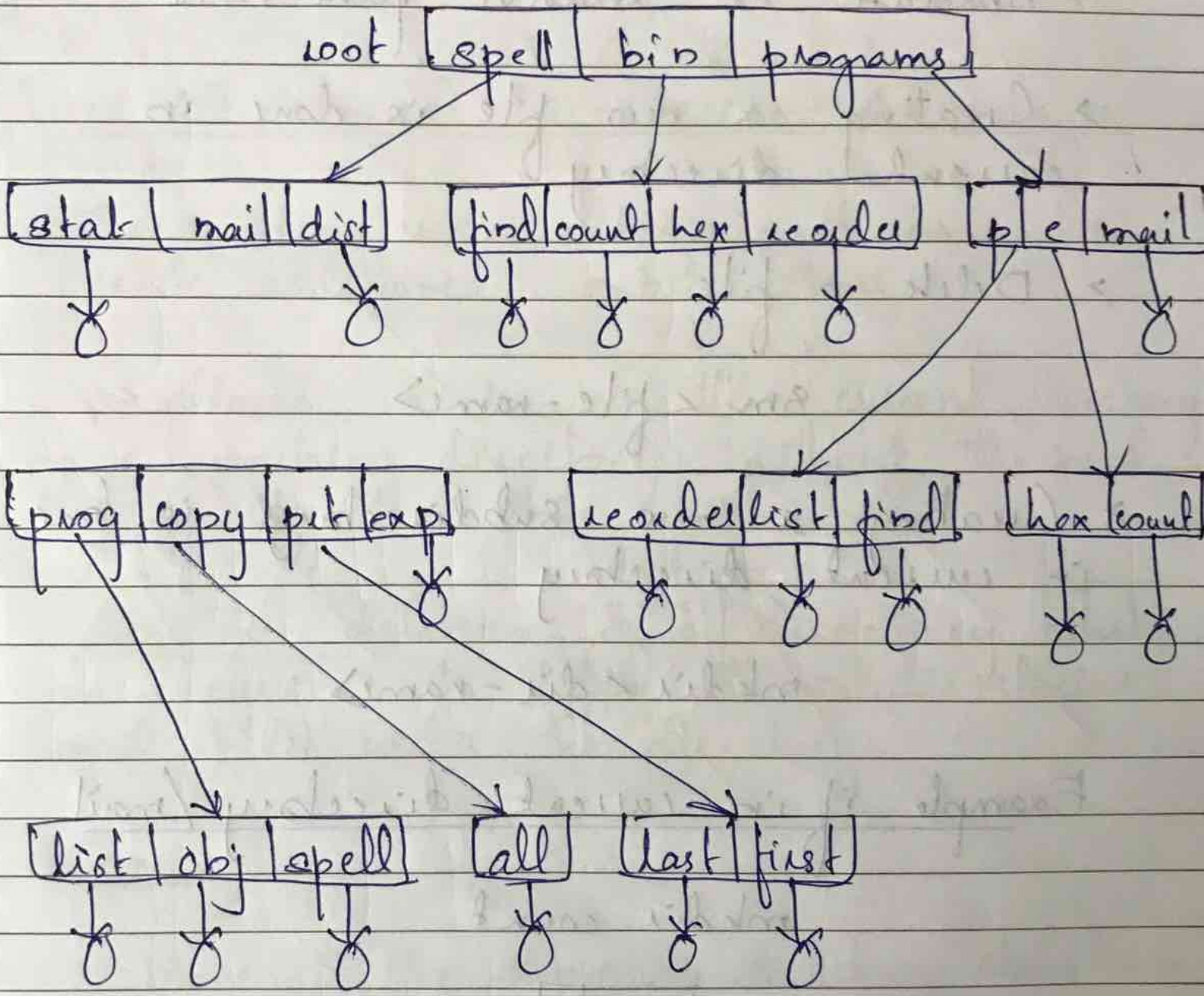


> Can have the same file name for different user

> Efficient searching

> No grouping capability

Tree - Structured Directories



directory - Sub directories - Relative path - Absolute path

user can create their own directory

classmate

Tree - Structured Directories

> Absolute or relative path name

> Creating a new file is done in current directory

> Delete a file

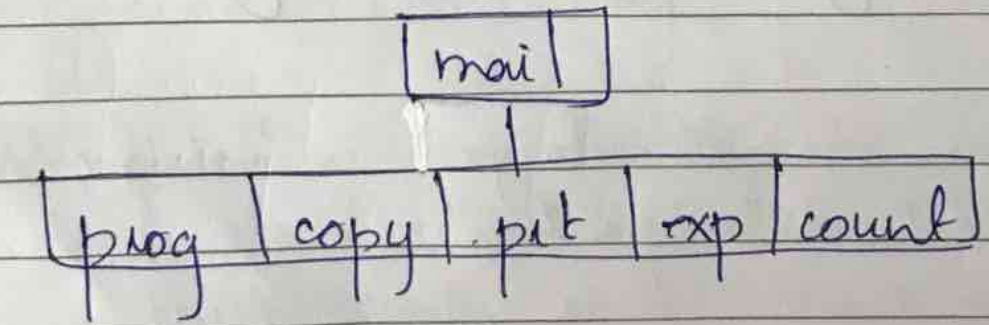
rm <file-name>

> Creating a new subdirectory is done in current directory

mkdir <dir-name>

Example if in current directory /mail

mkdir count



Deleting "mail" ⇒ deleting the entire subtree rooted by "mail".

Tree-Structured Directories - Advantages

> Tree-structure allows each user to create additional subdirectories

> allows us to use the current-directory as a working directory without the need to specify the whole path name.

> The deletion of a directory must be recursive to delete all subdirectories and files under the directory.

Cons

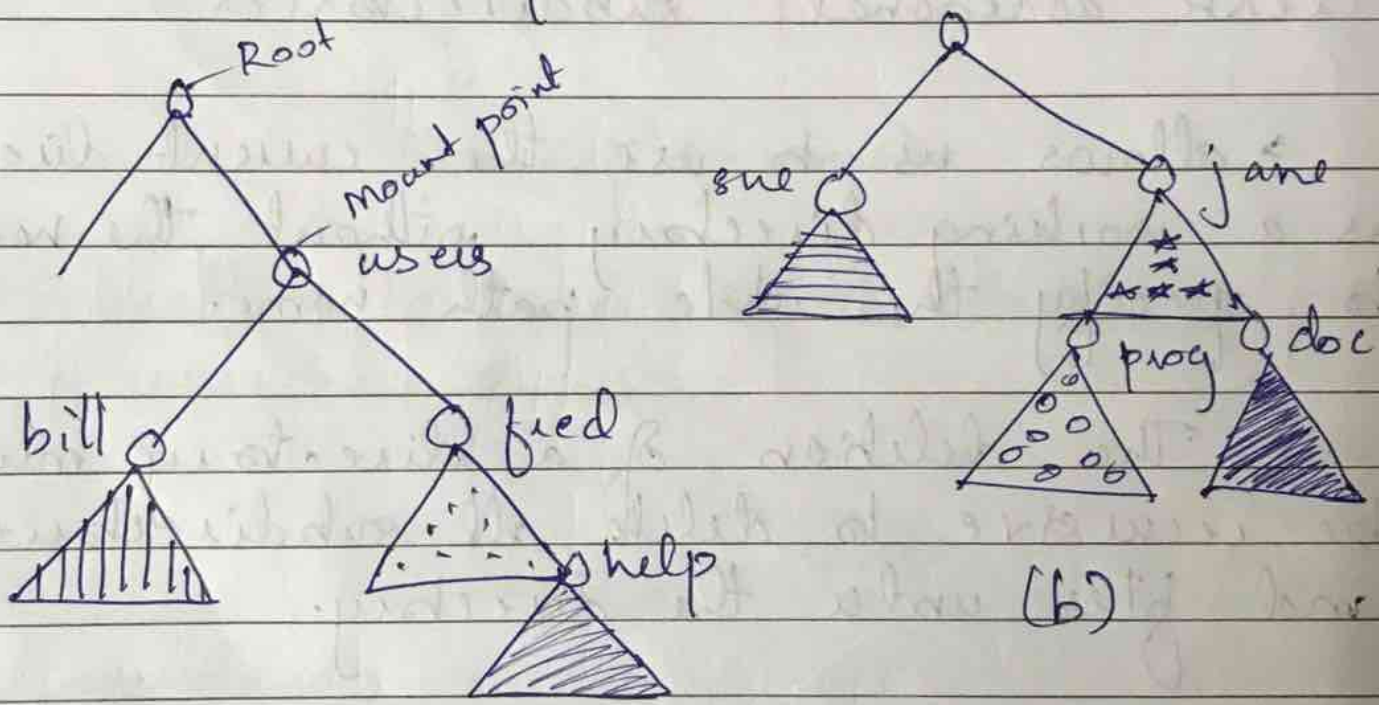
> None of the above 3 directory structures enable Sharing of files or subdirectories

> Adding additional links can result in a general graph or an acyclic graph.

File System Mounting

> A file system must be mounted before it can be accessed.

> A un-mounted file system is mounted at a mount point.



File sharing

> Sharing of files on multi-user systems is desirable

> Sharing may be done through a

protection schema

> On distributed systems, files may be shared across a network.

> Network File System (NFS) is a common distributed file-sharing method.

.

File Implementation

Protection

- File owner / creator should be able to control:
  - what can be done
  - by whom

- Types of access
  - Read
  - Write
  - Execute
  - Append
  - Delete
  - List

Access Lists and Groups

- Mode of access : read, write, execute
- Three classes of users

RWX

a) owner access 7 ⇒ 1 1 1

RWX

b) group access 6 ⇒ 110

RWX

c) public access 4 ⇒ 100

File-System Structure

- File structure
  - Logical storage unit
  - Collection of related information

- File system resides on secondary storage (disks)
  - Provided user interface to storage, mapping logical to physical.
  - Provided efficient and convenient access to disk by allowing data to be stored, located retrieved easily.

- Disk provides in-place rewrite and random access
  - I/O transfers performed in blocks of sectors (usually 512 bytes)

- File control block - storage structure consisting of information about a file.
- Device driver controls the physical device
- File system organized into layers

Layered file system

application programs

↓

logical file system → includes file-system structure & manages directory structure. A file control block (FCB) contains information about the file

↓

file-organization module → logical block to physical block translation and also manage the free block on the d

↓

basic file system → issue generic commands to device driver to read/write physical blocks on the disk eg! drive 3, cylinder 60, track 12, sector 10

↓

I/O control → Transfer block between memory and the disk. This deals with interrupt, device drivers

↓

devices

File-System Implementation

- We have system calls at the API level, but how do we implement their functions?
  - On-disk and in-memory structures

- On-Disk structure
  - Boot control block contains info. needed by system to boot OS from that volume.
    - Needed if volume contains OS, usually first block of volume.
  - Volume / Partition control block (superblock (UFS), master file table (NTFS) contains volume details
    - Total # of blocks, # of free blocks, block size, free block pointers or array
  - Directory structure organizes the files
  - Names and inode numbers (UFS), master file table (NTFS)
  - Per file FCB

- Logical file System manages metadata information
  - Translates file name into file number location by maintaining file control blocks
  - Directory management
  - Protection

- Layering useful for reducing complexity and redundancy, but adds overhead and can decrease
  - Logical layers can be implemented by any coding method according to OS designer.

FCB - maximum 64

A Typical File control Block

file permissions

file dates (create, access, write)

file owner, group, ACL

file size

file data blocks or pointers to file data blocks

In memory file system structure

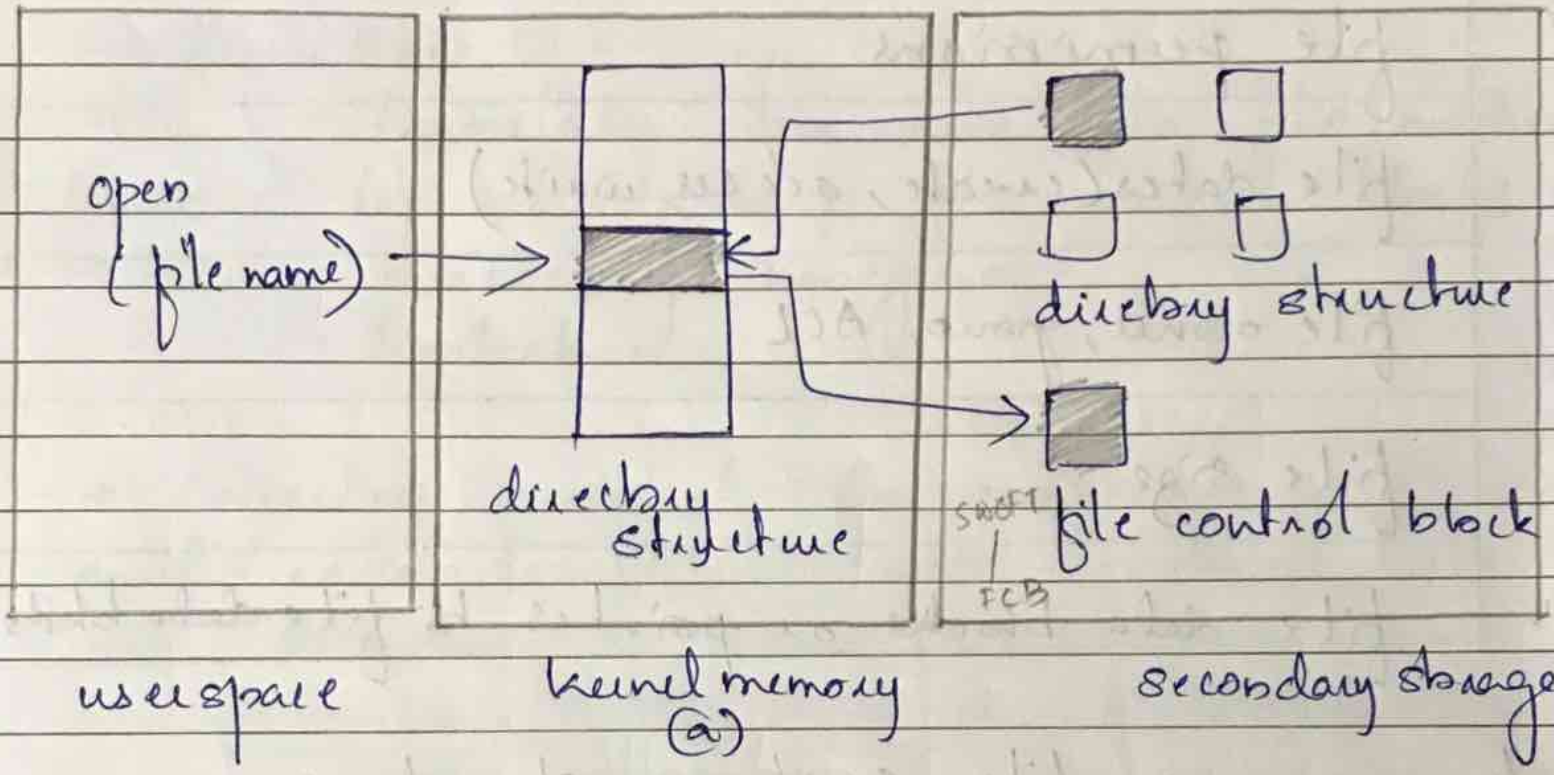
- In-memory partition table - It contains information about each mounted volume.
- In-memory directory structure - directory information of recently accessed directories.
- System wide open-file table - It contains the copy of FCB of each open file.
- Per-process open-file table - It contains information opened by that particular process and it maps with appropriate system wide open-file.

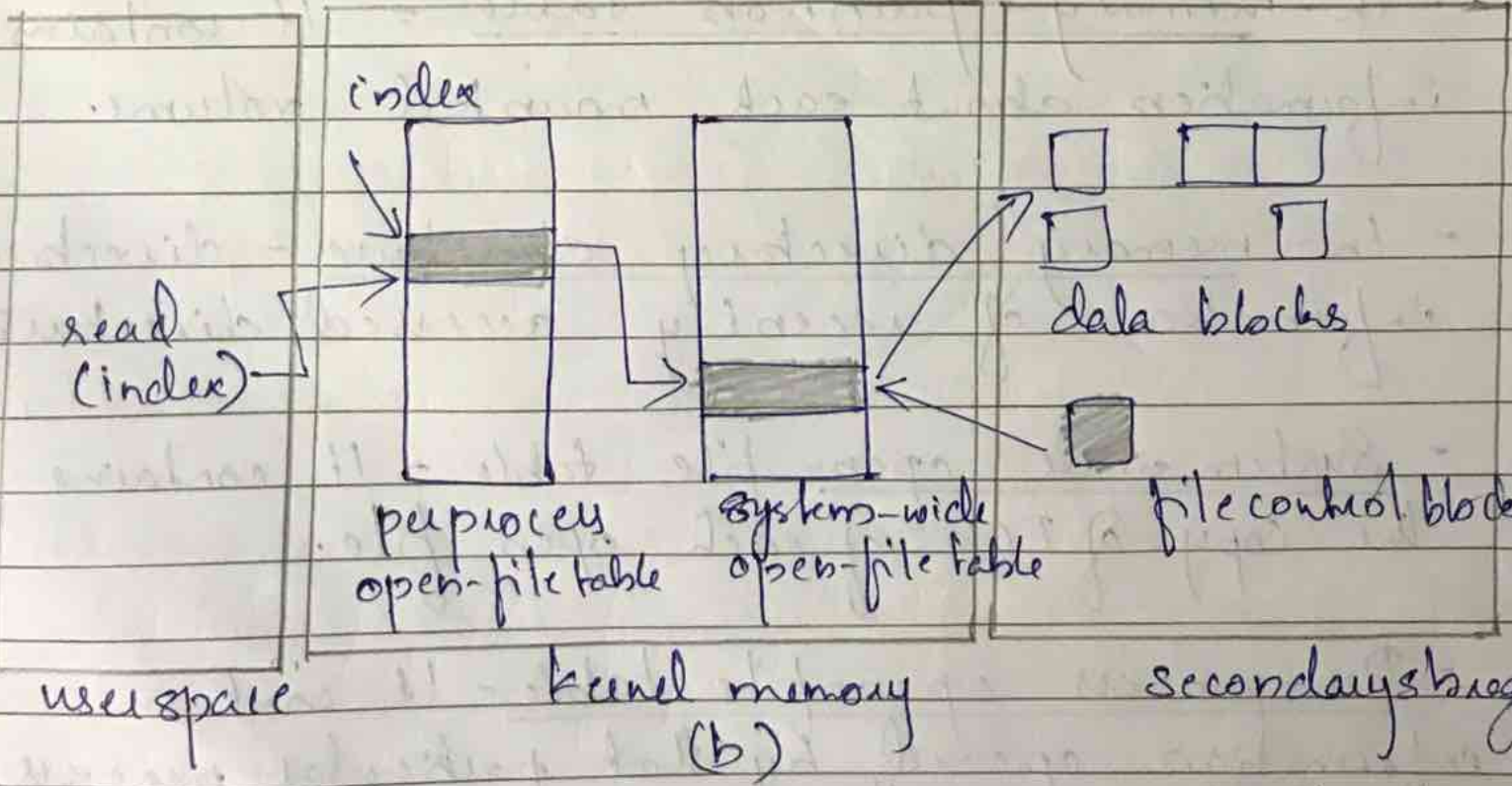
pointer of the entry in SWOFT

CLASSMATE

checks in kernal memory, checks there is entry or or not of FCB in SWOFT

In-memory File System Structures





Directory Implementation

- Linear list of file names with pointer to the data blocks.
  - simple to program
  - time-consuming to execute
- Hash table - linear list with hash data structure.
  - decreases directory search time.
  - collisions - situations where two file names hash to the same location
  - fixed size.

Hashing function

$H(x) = x \bmod$ no. of files

if 5 file, it will be 5

To create a file

→ Search dir to be sure that no existing file has the same name.

→ Add a new entry

↳ FCB

fi

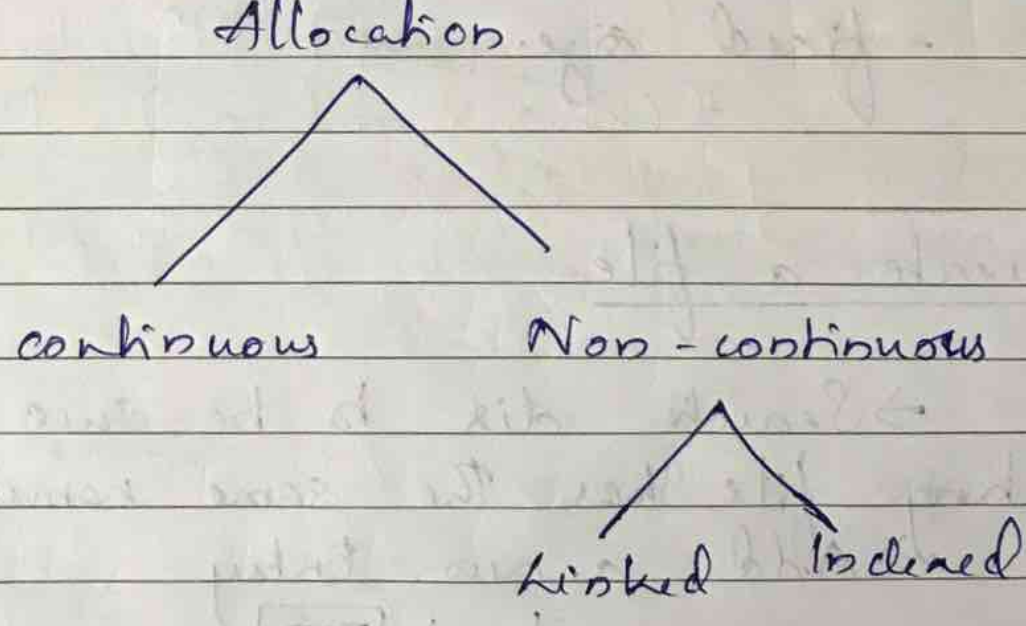
Allocation methods

> An allocation method refers to how disk blocks are allocated for files:

> Contiguous allocation

> Linked allocation } Non-Continuous

> Indexed allocation }



Why do you prefer allocation method?

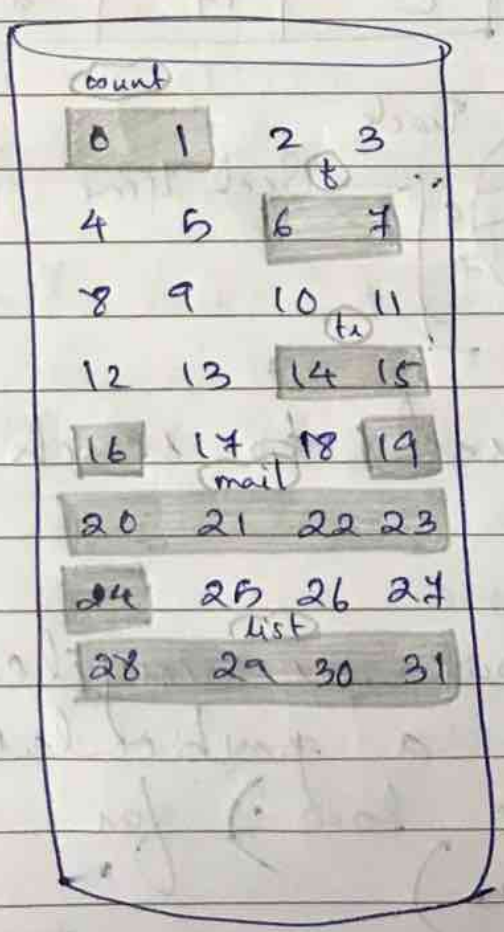
It is imp. for the ① utilization of disk space and ② access of data or access time.

disk - secondary storage.

Contiguous Allocation of Disk Space

Directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2



Continous Allocation :-

* When the logical blocks are allocated continously in disks. Does not need to start from zero all the time. From whereever it starts, it is allocated continously.
* Used for the utilization of disks space & for faster accss of data.

classmate

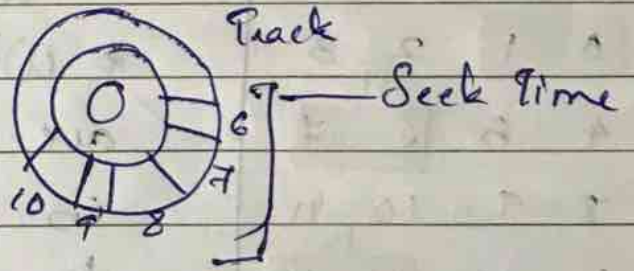
Advantages of contiguous allocation

(1) Simple (simple method, to understand)

(2) Excellent Read performance (wrt)
↳ Data can be fetched easily.

Directory

file	start	length
A	0	3
B	6	5
C	14	10



Seek time - Time required to reach a particular track.

* Seek time will be low or very less.
@ (once you reached a particular track then seek time will be low for contiguous allocation)

Disadvantages of contiguous allocation.

1. Internal fragmentation
2. External fragmentation
3. File cannot grow.

Internal fragmentation.

If we have a disk of 5kb & the file stored is of less capacity the left out storage remains unused.

External fragmentation.

We have free spaces, but don't have continous space to be allocated by the file. It is called as external fragmentation.

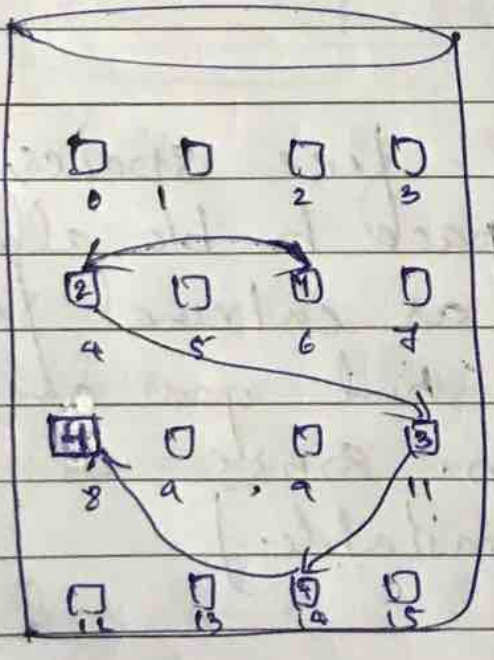
[you have free space but you are not able to use them. Since no continous memory available.]

Linked Allocation

→ Each file is a linked list of data & the pointer to the next data block.

→ File is stored randomly in a disk block.

→ At end, the pointer is set to null, which means that there are no more data blocks.



Directory

file	start	end
Jeep	6	8

- In this scheme, each file is a linked list of disk blocks which need not be contiguous.
- The disk blocks can be scattered anywhere on the disk.

The directory entry contains a pointer to the starting and the ending file block.

- Each block contains a pointer to the next block occupied by the file.
- The file 'jeep' in following image shows how the blocks are randomly distributed.
- The last block (8) contains -1 indicating a null pointer & doesnot point to any other block.

Advantage

- There is no external fragmentation & the file can grow.

Disadvantages

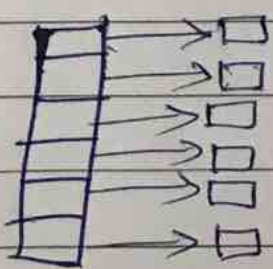
- Since we make use of pointers, separate memory is needed for storing the pointers
- Seek time is high / large seek time.

Indexed Allocation

Similar to index sequential access method.

- Bring all pointers together into the index block.

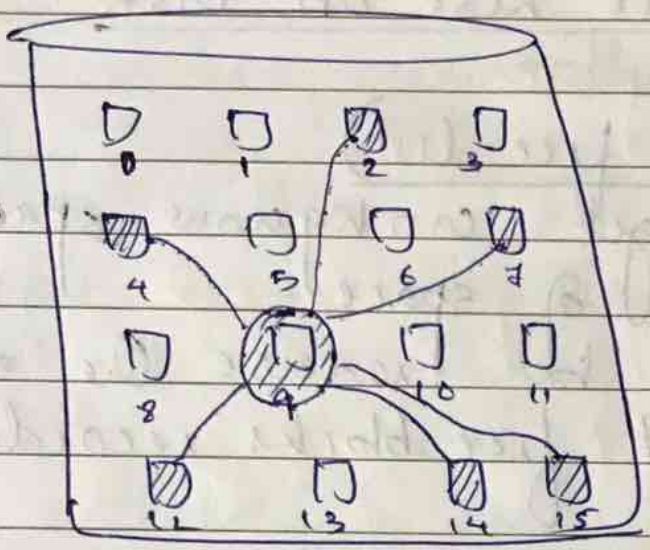
- Logical view →



index table

Directory

file	index block
jeep	9



Free Space Management

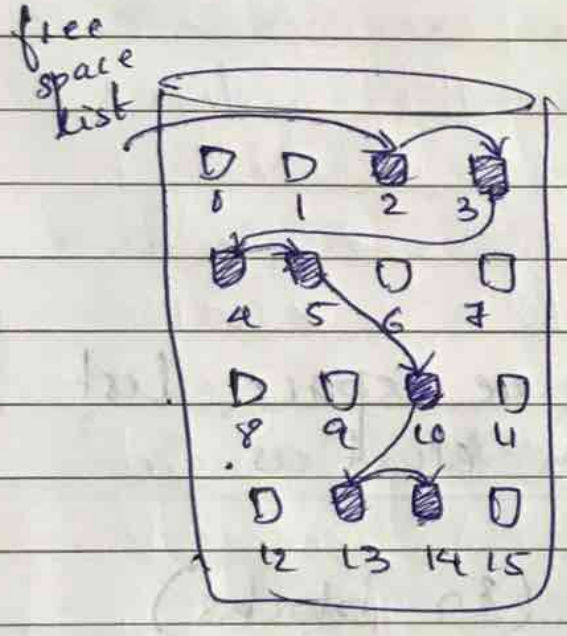
→ File system maintains free space list to track available blocks / clusters.

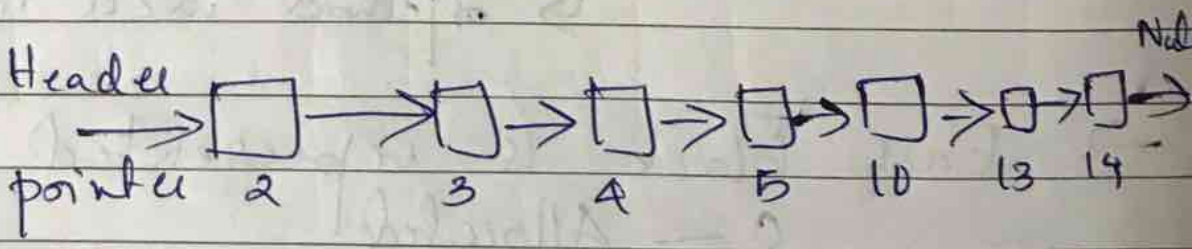
→ Bit vector or bit map (n blocks)
↳ method used in free space

→ Each block is represented by 1 bit.
0 — Allocated
1 — free

Linked Free Space List on Disk

- **Linked list (free list)**
  - Cannot get contiguous space easily
  - No waste of space
  - No need to traverse the entire list (if # free blocks recorded)





Linked free space on disk

→ Contains the pointer pointing to the first free space.

Grouping

This approach stores the address of the free blocks in the first free block. The first free block stores the address of some, say n free blocks. Out of these n blocks, the first n-1 blocks are actually free and the last block contains the address of next free n blocks.

Counting

→ Because space is frequently contiguously used and freed, with contiguous allocation, extents or clustering

- keep address of first free block and count of following free blocks.

- Free space list then has entries containing addresses and counts.

4/10/21

Linux Operating System

- Linux, computer operating system created in the early 1990s by Finnish software engineer Linus Torvalds and the Free Software Foundation (FSF)
- While still a student at the University of Helsinki, Torvalds started developing Linux to create a system similar to MINIX, a UNIX operating system.
- In 1991 he released version 0.02; Version 1.0 of the Linux kernel, the core of the operating system, was released in 1994.

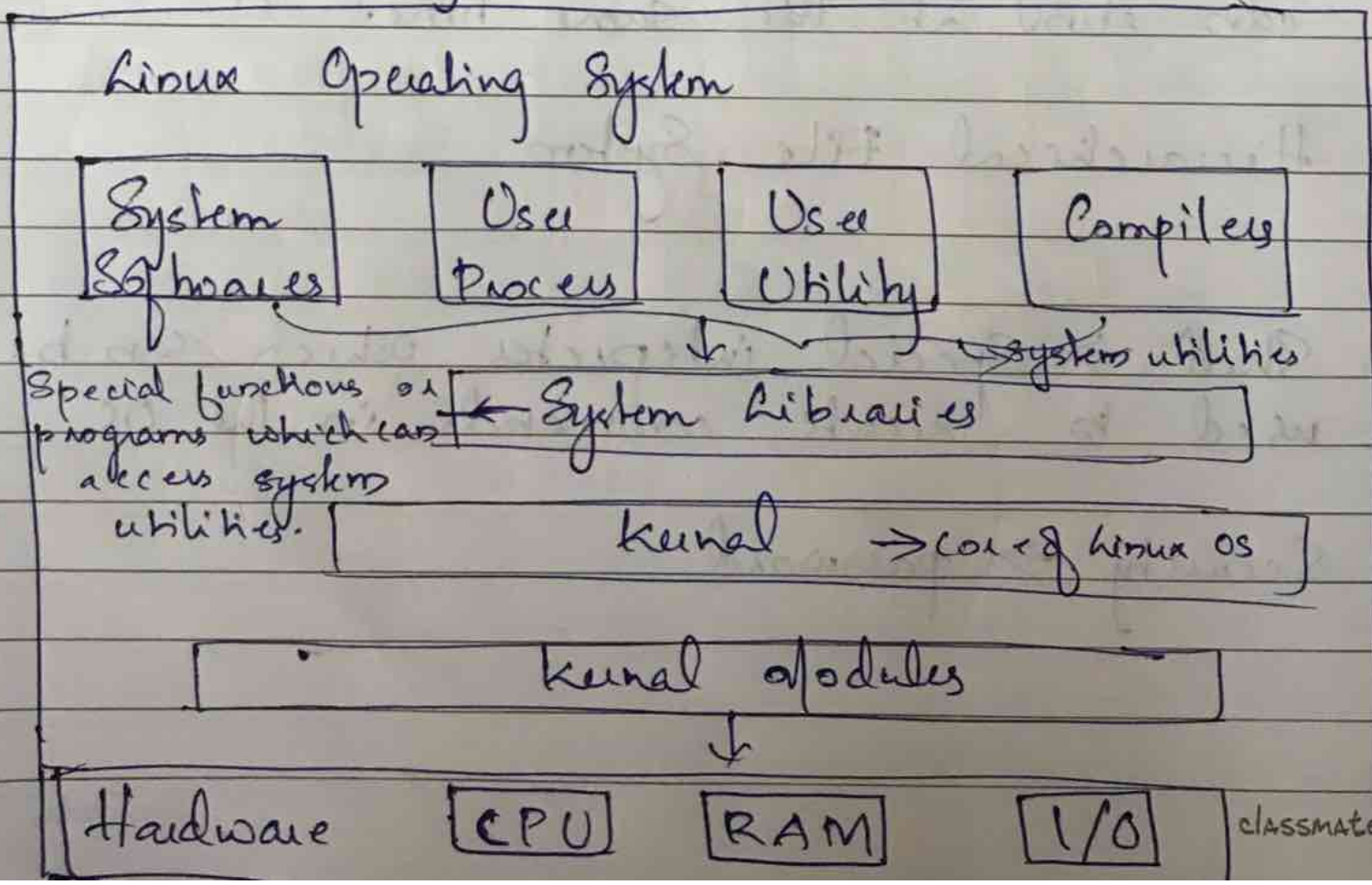
Features of Linux OS

- Portable → Different types of hardware in same way
- Open Source → Code is freely available
- Multi-User → Multiple user can access at the same time.
- Multiprogramming → Multiple applications can run at the same time.
- Hierarchical file System
- Shell → Special interpreter which can be used to execute commands in the OS
- Security → password

Some Linux Flavors

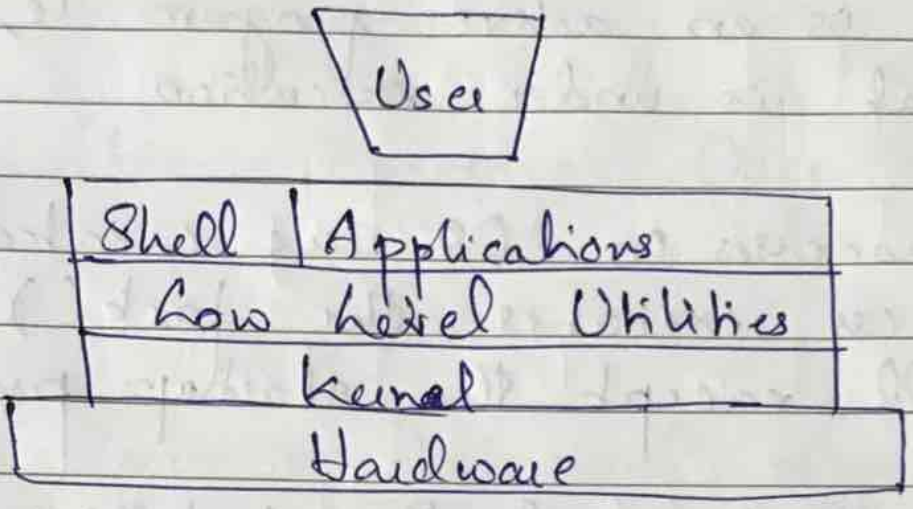
- offx Linux
- offanjalo
- Linux offint
- Ubuntu
- Debian
- Elementary OS
- Solus
- Zorin OS
- Fedora

Linux OS Layers



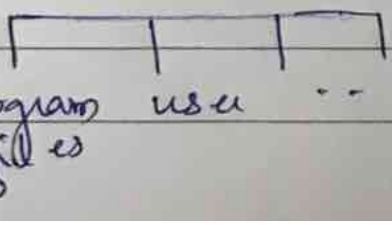
classmate

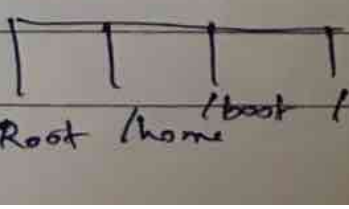
Shell in Linux / OS



- The shell is the command interpreter in an operating system.
- A Shell provides you with an interface to the Unix system.

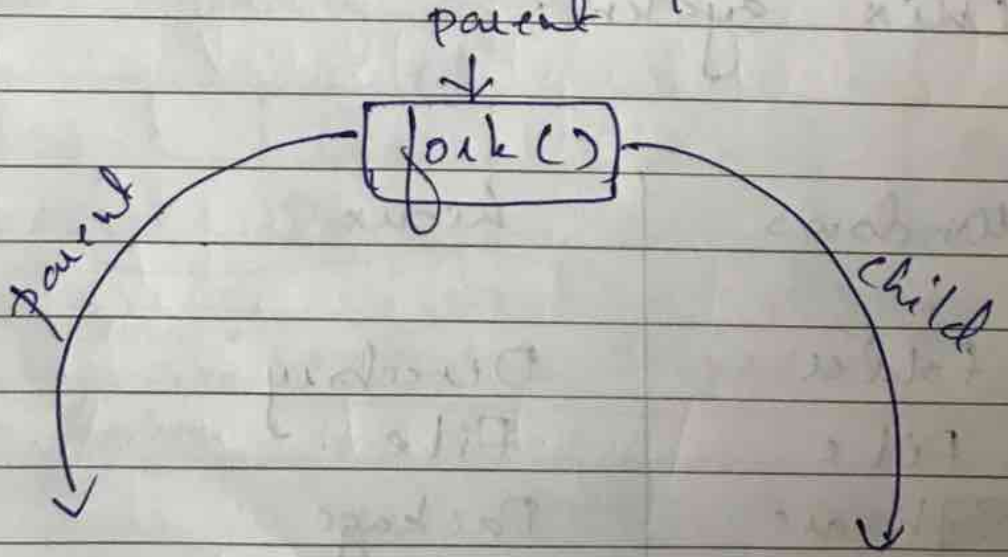
Windows	Linux
Folder	Directory
File	File
Software	Package
C:	C: / → Topmost level





Process: Parent and Child Processes

- A process is an active program ie, a program that is under execution
- All the processes in OS are created when a process executes the fork() system call except the startup process
- The process that used the fork() system call is the parent process.
- In other words, a parent process is one that creates a child process



Files and Directories

1. / - The Root Directory
2. /bin - Essential User Binaries
3. /boot - Static Boot Files
4. /cdrom - Historical mount Point for CD-ROMs
5. /dev - Device Files
6. /etc - Configuration Files
7. /home - Home Folders
8. /lib - Essential Shared Libraries
9. /lost+found - Recovered Files
10. /media - Removable Media
11. /mnt - Temporary mount Points

(12) /opt – Optional Packages

(13) /proc – Kernel & Process Files

(14) /root – Root Home Directory

(15) /run – Application State Files

(16) /sbin – System Administration Binaries

(17) /tmp – Temporary Files

(18) /usr – User Binaries & Read-Only Data

(19) /var – Variable Data Files

https://www.webmin.org/

(1) pwd
(2) mkdir Os
(3) cd Os
(4) touch fi.txt - Create a new file - touch
(5) echo "good morning"
(6) echo "this is a file with student names" > fi.txt
(7) cat fi.txt
↳ display content of file
(8) rm fi.txt
(9) rmdir
(10) date
(11) man

Basic Commands

1. pwd – to find out the path of the current working directory
2. cd – change directory
   - Syntax : cd <<directory-name>>
3. mkdir – make a new directory
   - Syntax : mkdir <<directory-name>>
4. rmdir – To remove a directory (the directory should be empty)
5. echo – print a message or value in screen
   - Syntax : echo <<message>>
6. ls – list directories and files
   - Options : ls -l, ls -s, ls -a

(7) rm - remove files / directory

- rm << file_name >> / rm -r << directory-name >>

(8) man - to print online reference manual

- Syntax : man << command_name >>

(9) cat - print content of a file OR concatenate two files

- Syntax : cat << file_name >>
- Syntax : cat << file1 >> << file2 >>

(10) touch - modify timestamp of a file

- Syntax : touch << filename >>

(11) date - to print current date and time

- Syntax : date.

5/10/21

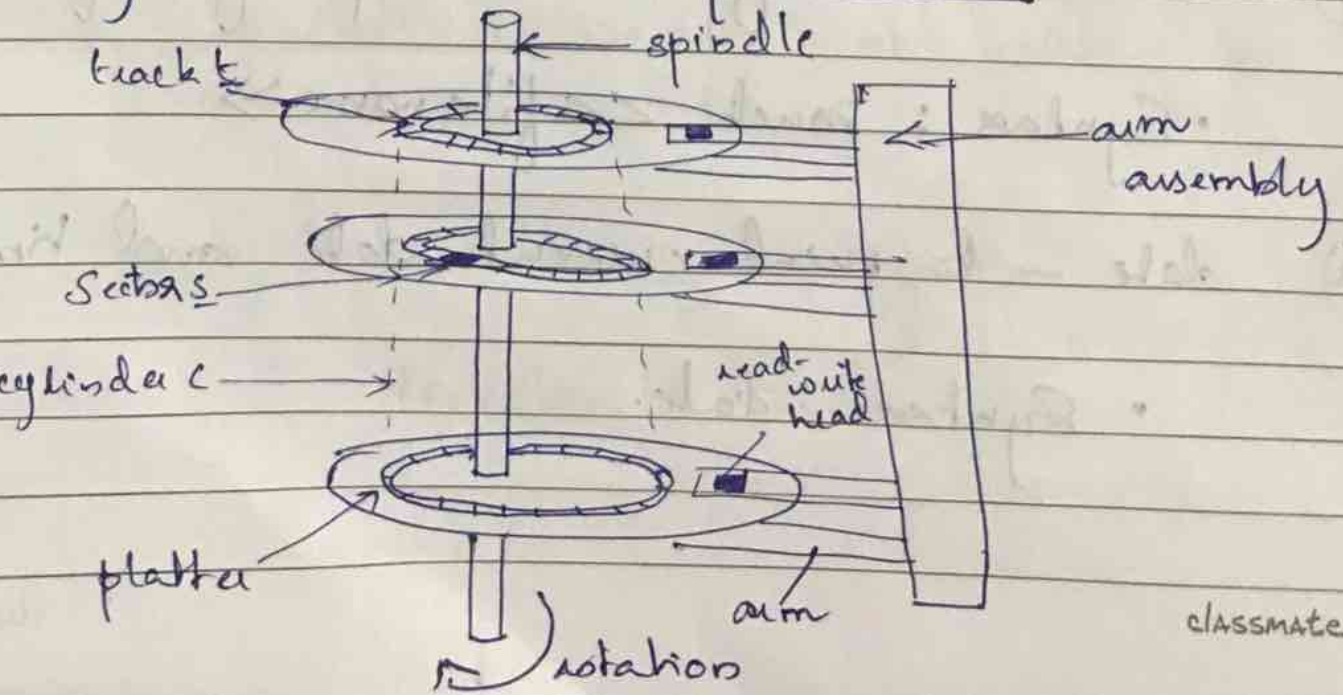
Disk Management

Mass - Storage Structure

> Magnetic Disks

- Relatively simple
- Disk platter has a flat circular shape
- Platter diameter can be between 1.8 to 5.25 inches
- Surfaces are covered with magnetic material
- Stores information by recording it magnetically on the platters.

Moving - head Disk Mechanism



classmate

> Magnetic Tapes

- Early Storage mechanism
- Is relatively permanent and can hold large quantities of data
- Access Time is slow compared with that of main memory and magnetic disks.
- Random access to magnetic tapes are very slower than random access to magnetic disks
- Not very useful for secondary storage.
  - Mainly used for backups, infrequently used data, or as a medium of data transfer.
- Storage - typically, 20 GB to 200 GB

Disk Attachment

> Computers access disk storage in two ways:

> Via Input - Output Ports (host - attached storage)
- Common on small systems
- Examples : hard disk drives, RAID arrays, CD, DVD, tapes, etc..

classmate

> Via a Remote Host in a Distributed File System (network-attached storage)

- Special purpose storage system accessed over a data network.
- Clients access NAS via remote procedure call interface
- A special case - Storage Area Network (SAN)

> Seek Time: Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or write. So the disk scheduling algorithm that gives minimum average seek time is better.

> Rotational Latency: Rotational latency is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads. So the disk scheduling algorithm that gives minimum rotational latency is better.

> Transfer Time: Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred.

The First Commercial Disk Drive

1956
IBM RAMDAC Computer included the IBM model 350 disk storage system.

5M (7 bit) characters
50 x 24" platters
Access time = < 1 second

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth.

- Access time has two major components
  1. Seek time is the time for the

disk are to move the heads to the cylinder containing the desired sector.

② Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head.

- Minimize seek time
- Seek time ≈ seek distance

> If the desired disk drive & controller are available then the request can be served immediately

> If the drive or controller is busy, any new requests for service will be placed in the queue.

> When one request is completed the operating system chooses which pending request to service next.

> How does OS make this choice?
↳ Ans: Disk Scheduling Algorithms

classmate

Disk Scheduling Algorithms

> FCFS (First Come First Served) Schedulin

> SSTF (Shortest Seek Time First) Scheduling

> SCAN Schedaling

> C-SCAN Scheduling

> LOOK Scheduling

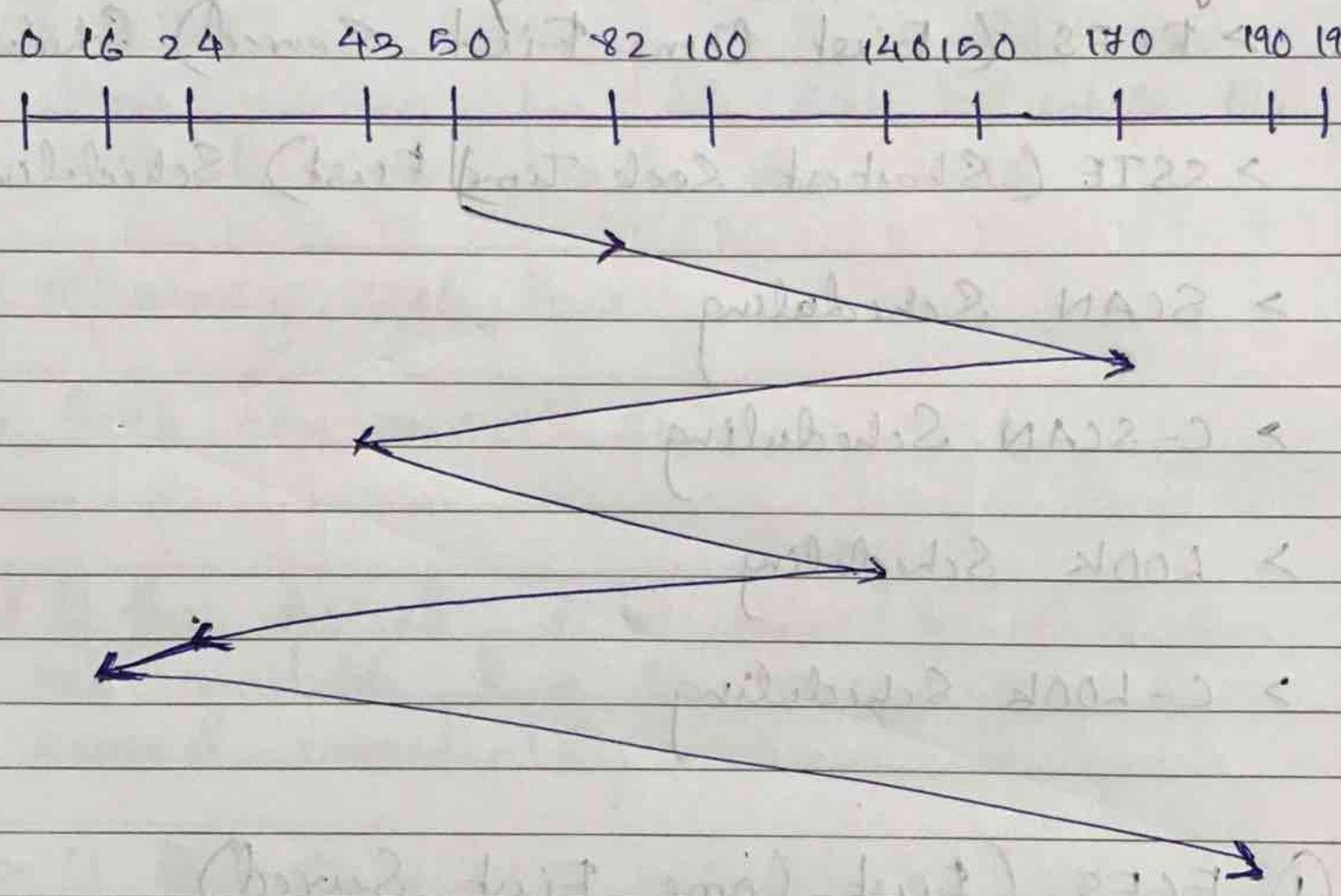
> C-LOOK Scheduling

① FCFS (First Come First Served)

- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue (0-199)
- 82, 170, 43, 140, 24, 16, 190
- Head pointer 50
- Calculate the total no. of track movement by R/W head

82, 170, 43, 140, 24, 16, 190

illustration show total head movement of 642



So total seek time:

(82-50) + (170-82) + (170-43) + (140-43)
+ (140-24) + (24-16) + (190-16)

= 642 //

classmate

H.W 2

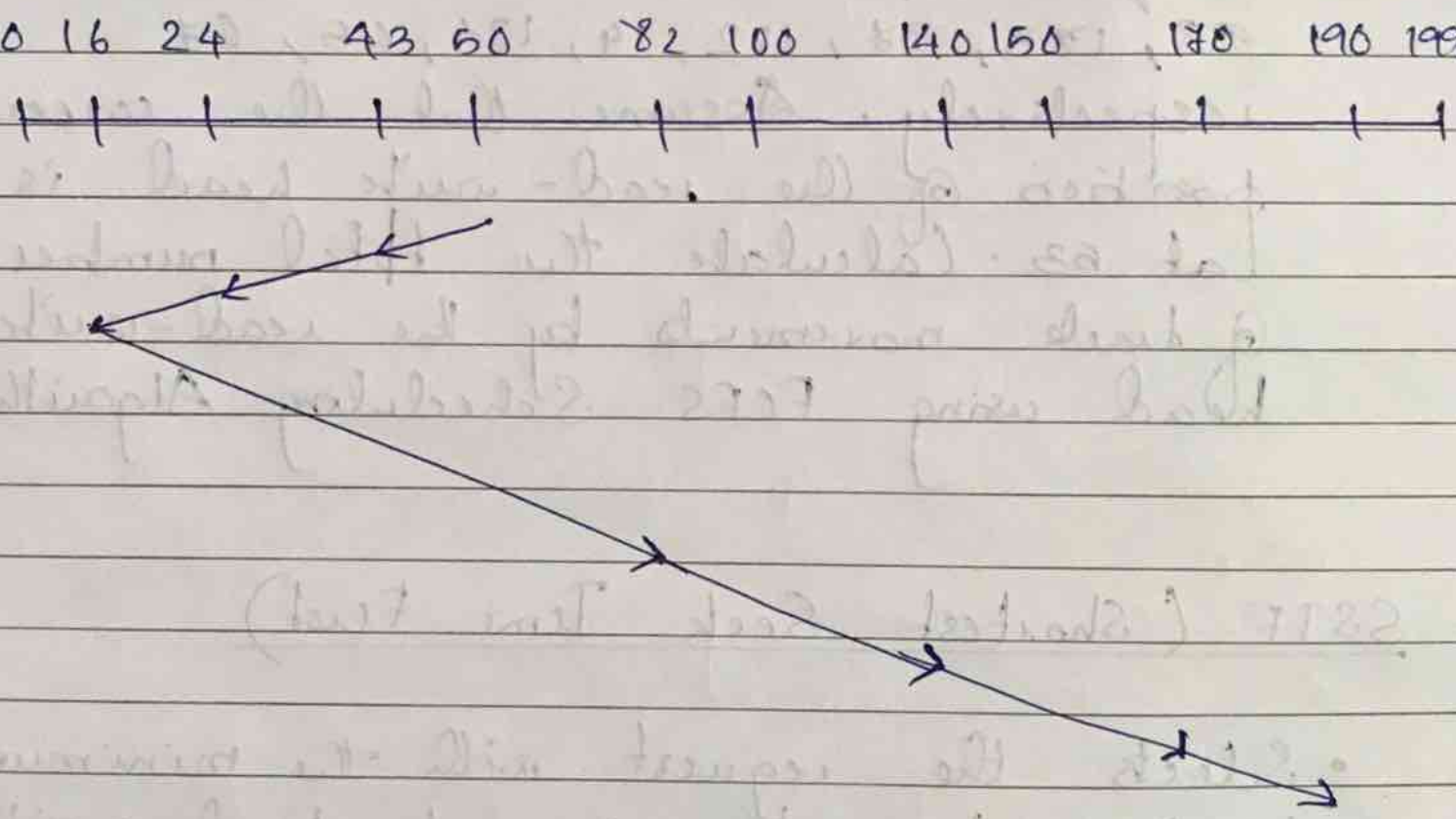
FCFS, SSTF, SCAN,
(Practice Question)

Q. A disk contains 200 tracks (0-199). Request queue contains track numbers 98, 183, 37, 122, 14, 124, 65, 67 respectively. Assume that the current position of the read-write head is at 53. Calculate the total number of track movements by the read-write head using FCFS Scheduling Algorithm.

SSTF (Shortest Seek Time First)

- Selects the request with the minimum seek time from the current head position.
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.

A disk contains 200 tracks. Request queue contains 82, 170, 43, 140, 24, 16, 190. Head pointer is at 50.



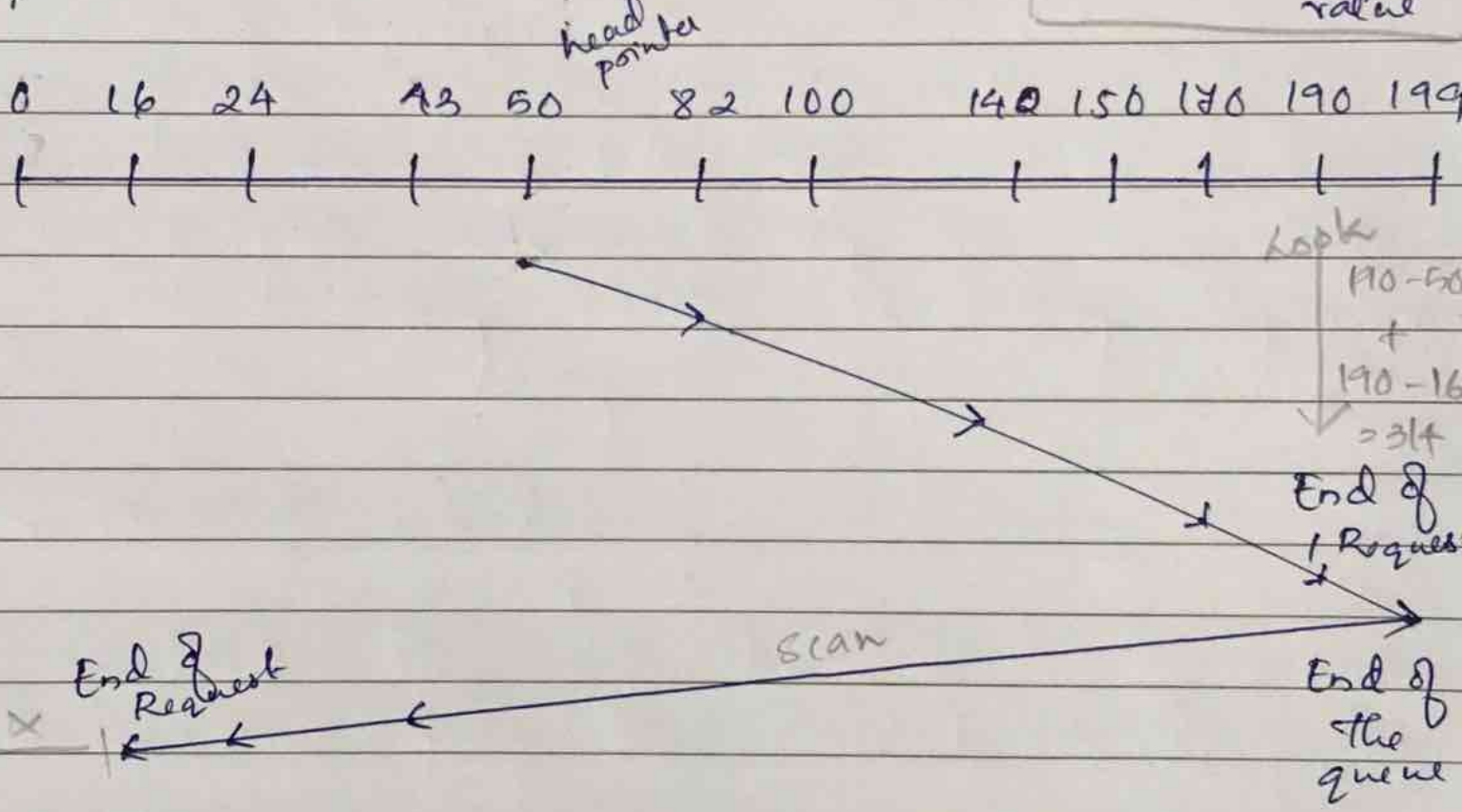
(50-43) + (43-24) + (24-16) + (82-16) + (140-82) + (170-140) + (190-170) = 208

Illustration shows total head movement of 208

SCAN

A disk contains 200 tracks. Request queue contains 82, 170, 43, 140, 24, 16, 190. Head pointer is at 50

Direction – lowest → highest value



$= (199 - 50) + (199 - 16) = 332$