PROF. ANAND N GHARU
PVGCOE NASHIK
8087777708

Name: Gharu Anand
Assistant Professor
Computer Dept
PVGCOE, Nasik
SPOS
TE COMPUTER

UNIT-6

8. I/P & O/P File system

1) I/O devices :-

- Block devices - store fixed size block, Each block has unique ID.
- Character devices

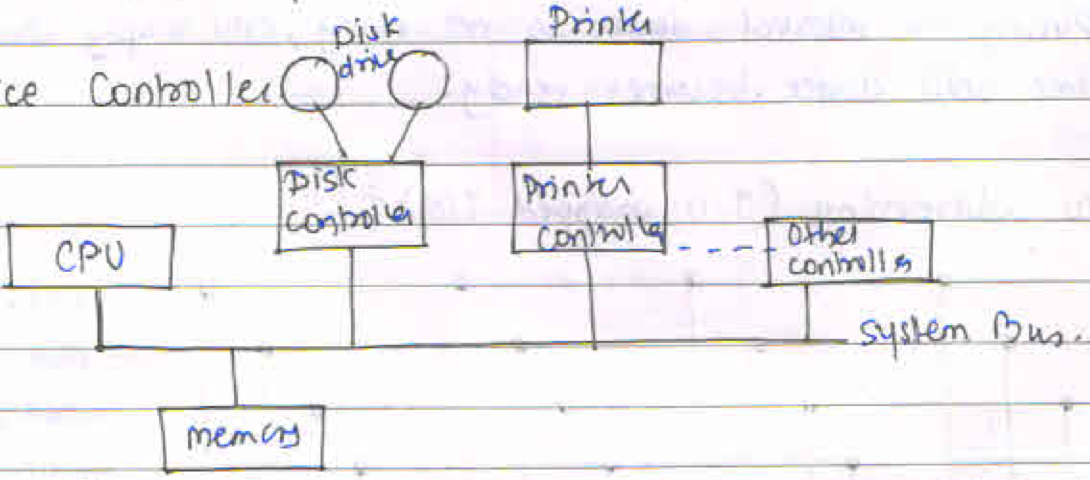
* Block device

- it stores infor in fixed-sized block.
- Each block has unique ID.
- Basic unit of R/W in block device is a block.
- block can be accessed independantly.
- Disk is block addressed device.
- Any block can be accessed in disk. respective of current position of R/W head.
- Similarly, a tape or a CD is block device.

* Character Device :-

- it accepts or deliver stream of characters.
- There no concept of blocking..
- Terminal, printer, n/w interfaces, keybord etc are character device.

* Device Controller



Single Bus Connecting CPU, memory, controller & I/O devices

* Tech for performing I/O (org of I/O function)

Tech of DMA mode :-

- programmed I/P - O/P
- Interrupt driven I/P-O/P.
- Direct Memory Access

1) programmed I/O :-

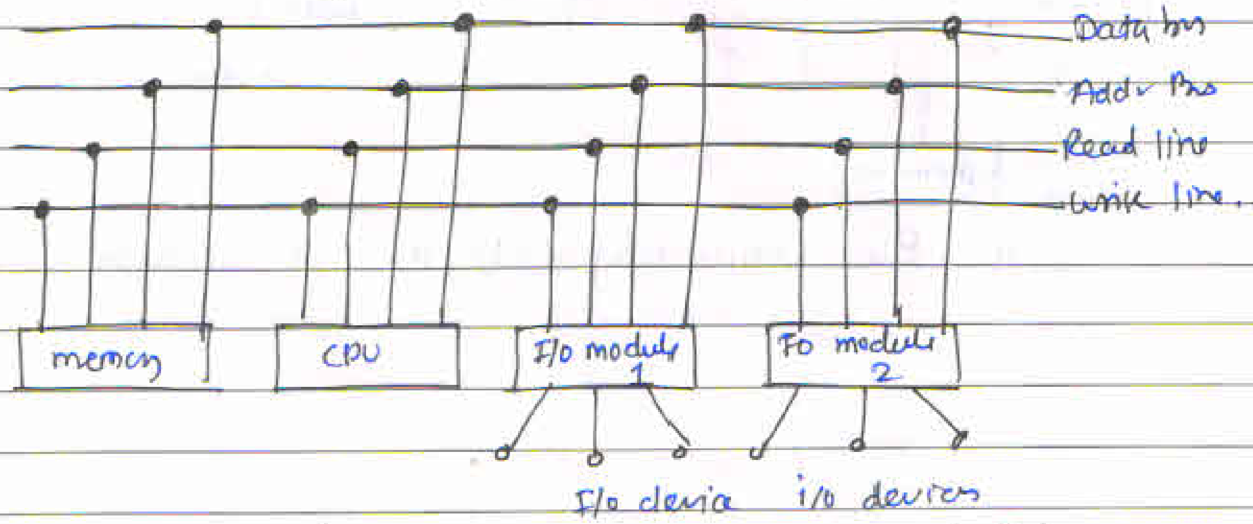
- it is useful methode for comp where h/w cost is minimized.
- entire i/o is handled by CPU with help of small S/w system. (sys s/w / device driver) w/o any addiHonal h/w.
- It is based on concept of Busy waiting, Befor I/O oper performed, CPU checks the status of I/O.
- If I/O is not ready, CPU waits in loop for I/O device to become ready.

CPU performs foll. step.

1) Read I/O device status bit
2) Test the status bit to determine if the device is ready to begin data transfer oper.
3) If devices is not ready, return to step-1, otherwise proceed with data transfer.

During the interval, device is not ready, CPU simply waste it's time until device becomes ready.

2) I/O addressing (I/O mapped I/O) :-



Structure of memory mapped I/O.

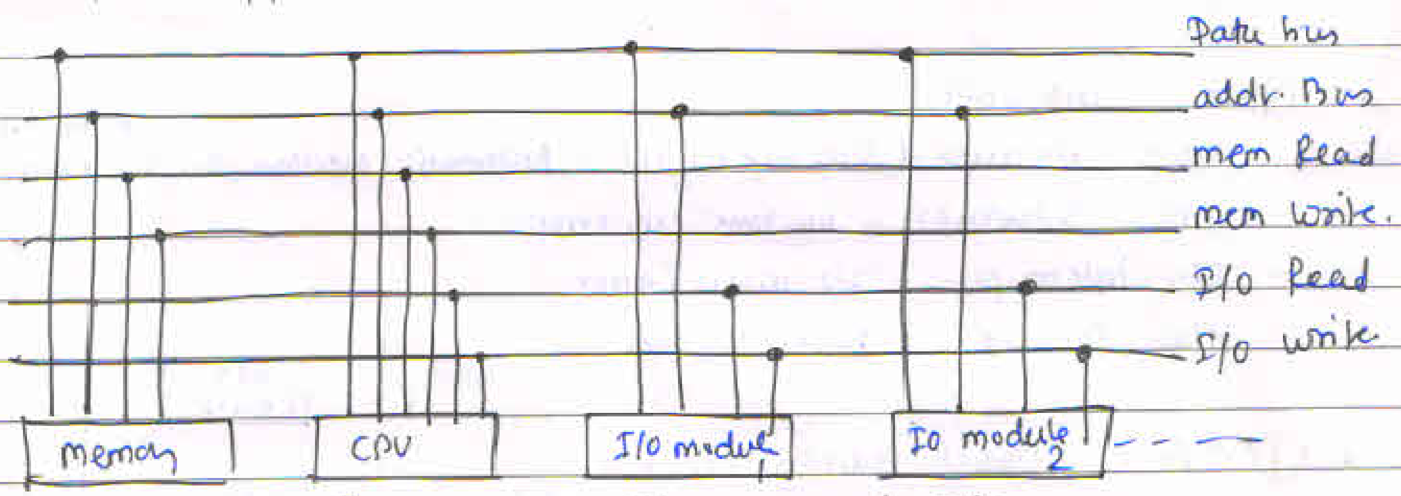
* Memory mapped I/O.
- In this, there is single addr. space for memory location and I/O devices.
- processor treates status & data reg. for seperate memory location. It is part of I/O device.
- processors uses same mem. loc instruction to access both mem. & I/O devices
- with mem mapped I/O, single R/W line are needed on the bus
- Read line is activated during transfer of data from memory to CPU.

e.g mov AX, x [AX ← x]

- write line is activated during transfer of data from CPU to memory

e.g mov x, AX [x ← AX].

* I/O mapped I/O :



Structure of I/O Mapped I/O.

- There are seperate control line for mem & I/O devices
- A mem refer. instruction does not affect on I/O devices.
- There are seperate address space for mem and I/O devices. An I/O device & mem loc can have same address.
- There are seperate instruction for memory Read & write.
- mem. Read instruction - mov AX, x
- mem write instruction - mov x, AX
- I/O read - IN AL, 300H
- I/O write - OUT 300H, AL.

3) I/T driven I/O.

÷ Major drawback of prog I/O is busy waiting.
- speed of I/O devices is much slower in comparison to that of CPU. since, in programmed I/O, CPU has repeatedly check whether a device is free, the performance of CPU goes down.

One solution could be ÷
- CPU switches to some other prog w/o waiting for I/O. device to complete or to become free.
- when devices becomes free, it informs backs the CPU thro. mechanism, known as interrupt.

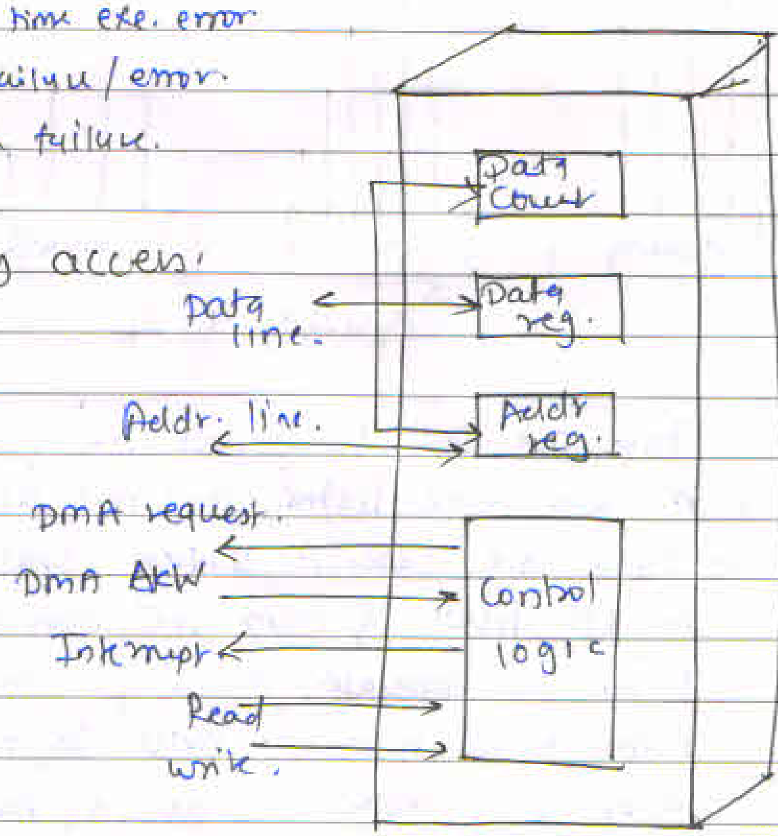
e.g. interrupt occurs, push context of current prog.

* Interrupt Processing ÷
already studied

* compare Subroutines of ISR.

* Types of Interrupt :
- prog. interrupt (S/W interrupt) - Arithmatic overflow, Divide by zero, mem. violate
- Timer Interrupt - on time exe. error
- I/O interrupt. - I/O failure / error.
- H/W failure. - power failure.

* DMA ÷ (Direct memory access.



Typical DMA Block diagram.

* File operation ÷
* Types of file ÷ regular, Directories, char special, Block special

PROF. ANAND N GHARU

* Write short notes on RAID.

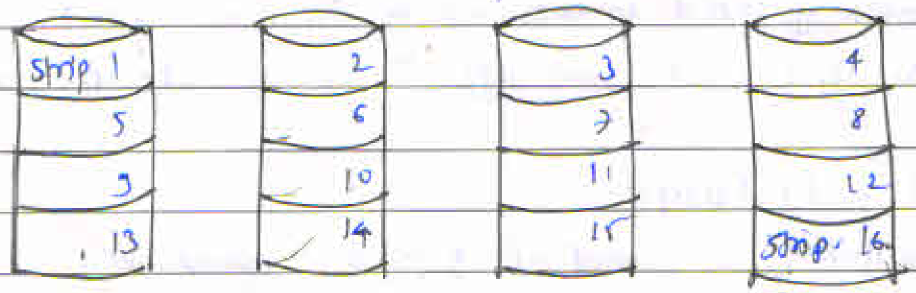
- Redundant Array of Independent Disks
- Magnetic disk has several drawback.
  1) They have relatively low transfer rate.
  2) Their electromagnetic design makes them prawn to fault.
- Data transfer rate can be increased by using an array of small disk unit. it operates ||ely.
- In addition, add^n of redundant disk unit store parity information can guarantee data recovery in case of disk failure.
- The idea behind RAID is to distribute the stored data over a set of disks configured to appear like a single large disk.
- The data can be distributed in various ways referred as RAID level 0-6.

The Basic characteristics of RAID disks are:
1) A set of physical disk appears as a single logical disk to OS.
2) Data are distributed across the physical disks.
3) In case of failure of disk, the parity info^n that is kept on redundant disk is used to recover the data.

The Features of these RAID level are:
1) RAID level 0 (Non-redundant)
2) RAID level 1 (Mirrored)
3) RAID level 2 (Redundancy thro. hamming Code)
4) RAID level 3 (Bit-Interleaved Parity)
5) RAID level 4 (Block-level Parity)
6) RAID level 5 (Block-level Distributed Parity)
7) RAID level 6. (Dual Redundancy).

PROF. ANAND N GHARU

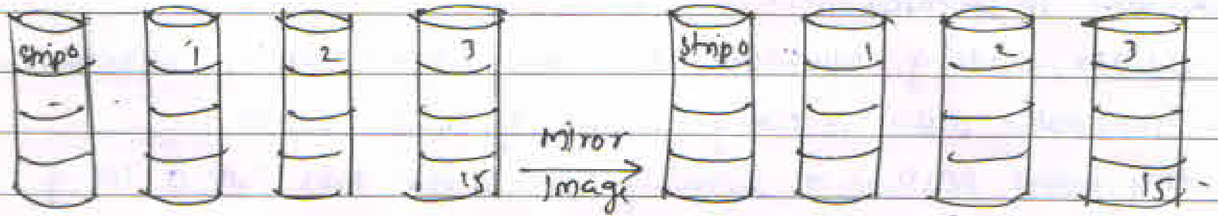
1) RAID level 0: It distribute data over an array of disks in the form of strips. These strips are stored in interleaved fashion.
   - For high data transfer capacity, strip size should be small.
   - it does not provide data recovery in case of disk failure.



RAID 0 (Non-Redundant)

2) RAID 1 : (Mirrored) :

- mirroring which keep a duplicate copy of data
- Any read request can be serviced by any of 2 disks. This may reduce seek time.
- Recovery from failure is done using mirrored disk.
- It provides real time backup.



RAID level 1 (Mirrored)

3) RAID 2 :- (Redundancy thro. hamming Code)

- it uses ||el access tech.
- Hamming error correction code is used that correct single bit error.
- it is used when data error are high.

4) RAID 3 : (Bit Interleaved Parity)

- It requires only single redundant disk, which is used as parity disk.
- parity disk can be used for reconstru of data in case of disk failure.
- As data strips are small, it may achieve very high data transfer rate

5) RAID 4 :- (Block-level Parity)

- it uses independant access tech, where each of these physical disk may be accessed independantly.
- Data strip is of larger size. & bit by bit parity strip is created for bit of strip of each disk
  - parity bit is stored in seperate disk.
  - Every write oper need updation of parity bit.
  - it is bottleneck as all I/O requests use it.

6) RAID 5 - Block level Distributed Parity.

- In this, bottleneck of level 4 is avoided by distributing parity strips
- It is useful for appli where very high I/O reques rate are needed

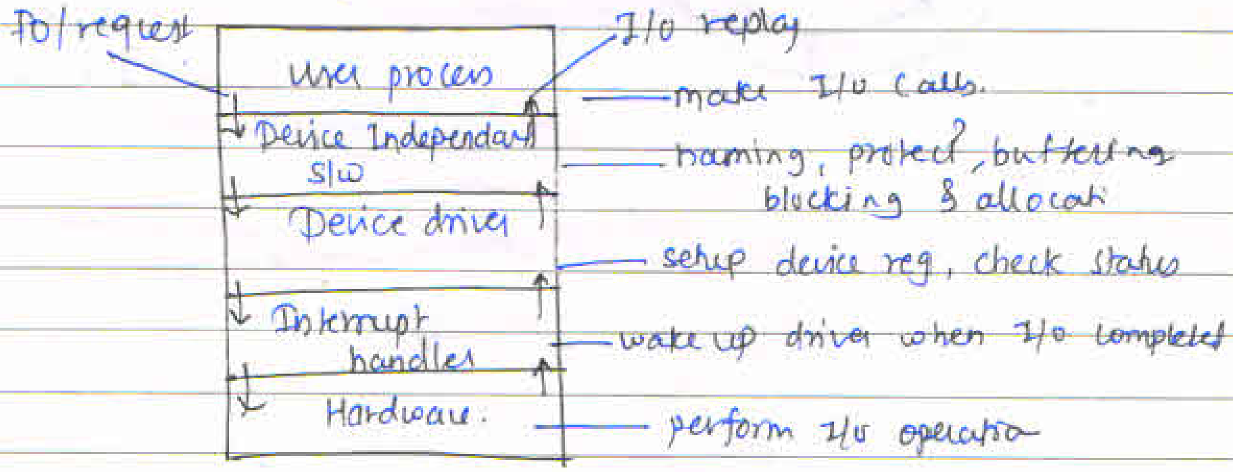
7) RAID 6 - (Dual Redundancy)

- 2 seperate parity calculation are carried out & stored in seperate block on diff disk.
- it provides extreamely high data reliability.

- DMA increases speed of I/O transfer by eliminating the role played by CPU in such oper.
- When large amount of data is to be transfered from CPU. a DMA module can be used.
- DMA operates in foll ways.
  1) when I/O requested, the CPU instruct the DMA module about the oper by providing the infor. -
     1) which oper is to be performed (R/W)
     2) starting loc in memory where the infor will be read or written.
     3) No of word to be written or read.
- The DMA module transfer the requested block byte by byte directly to memory w/o the intervention of CPU.
- On comple of request, DMA module send interrupt signal to the CPU.
- In DMA mode, the CPU intervention is limited at the beginning and end of the transfer.
- while DMA is busy in data transfer, CPU can execute another program.

* DMA Data transfer mode:-
- DMA Block transfer — HDD, magnetic disk
- Cycle stealing mode — I/O, DMA mode.
- Transparent DMA. — only unused sys. bus is used by DMA. don't distrub executing instn

* I/O software layer.



* I/O Software Layers :-

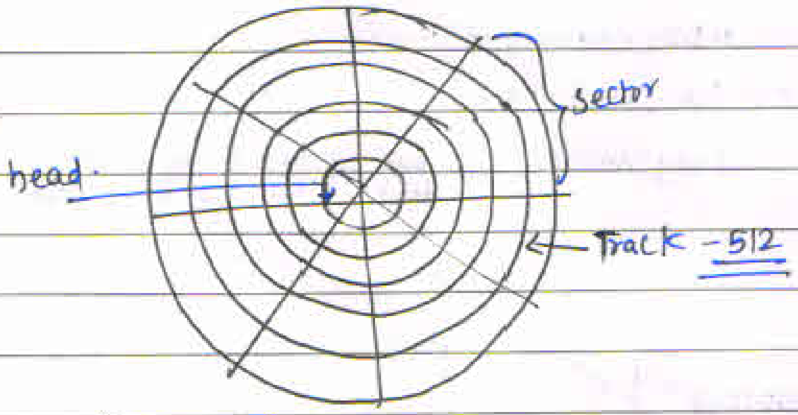
- lower layer interact with h/w & upper layer interact with users.
- I/O error is handled by I/O software.
- Interrupt are handled by I/O software.
- I/n I/O, read cmd the prog is automatically suspended until the data are available in the buffer.
- Schedulling of I/O devices is handled by I/O software.

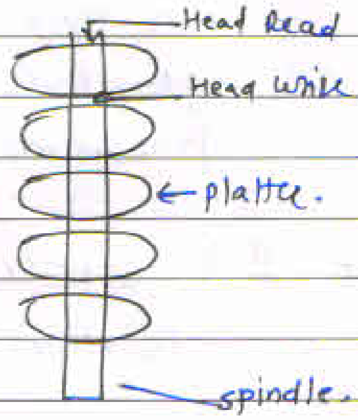
many devices like disk, can be used by many users at same time.

- many devices are dedicated. e.g. printer.
- The goal of I/O s/w can be achieved by structuring the I/O s/w in 4 layers.
- Interrupt Handler
- Device Driver
- Device Independant OS S/w
- User level s/w.

* Magnetic Disk :-

- Fixed head / movable head disks.
- sides
- platters
- Head Mechanism.
- Seek time
- Latency





* File Operation :-

- Creating a file.
- Opening a file
- Renaming
- Reading data from file
- writing data to file.
- Random access of record
- Getting attribute of file
- Setting attribute of file.
- Appending data to file.
- Closing a file.

Before performing any operation, file must be created on secondary storage.

- After creation of file, it should be either opened in read or write mode.
- We can read particular record or complete file.
- New record can be added to existing file.
- The record may be deleted if it no longer used

* File Types :-

- Regular files
- Directories
- Character Special file
- Block Special file.

1) Regular file :-

- regular files are either ascii or binary files. ascii files are also known as text files, it contain visible char.
- we can content of file on monitor or perform manipulation in file.
- e.g. of text file
  1) C-prog. file.
  2) A file containing a letter or text book.
- Text ~~book~~ consist of -
  alphabets or digit

e.g. docx. file, pdf file. etc

PROF. ANAND N GHARU

* File Operation :-

- Creating a file
- Opening a file
- Renaming
- Reading data from file
- writing data to file.
- Random access of record
- Getting attribute of file
- Setting attribute of file.
- Appending data to file.
- Closing a file.

Before performing any operⁿ, file must be created on secondary storage.

- After creation of file, it should be either opened in read or write mode.
- We can read particular record or complete file.
- New record can be added to existing file.
- The record may be deleted if it no longer used

* File Types :-

- Regular files
- Directories
- Character Special file
- Block Special file.

1) Regular file :-

- regular files are either ascii or binary files.
  ascii files are also known as text files, it contain visible char.
- we can content of file on monitor or perform manipulation in file.
- e.g. of text file
  1) C-prog. file.
  2) A file containing a letter or text book.
- Text ~~book~~ consist of -
  alphabets or digit
  e.g. docx. file, pdf file. etc

2] Directories:

- Directories are system files for maintaining the structure g the file system.

e.g

C:\ program files\ java\ ddk 1.8 ... ..

3] character special files:-

- These are sys. files. They are related to I/O and used to model serial I/O devices such as terminals, printer & n/ws.

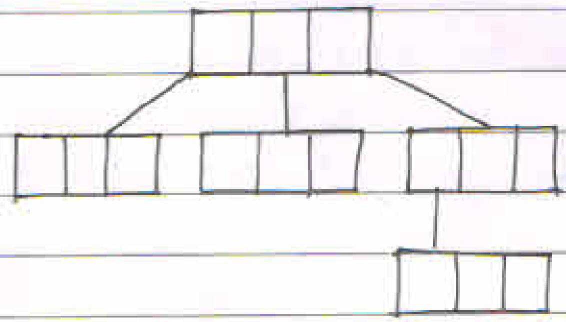
4] Block special files:-

- Blocks files are used to model disk.

* File structure:-

- There are 3 common file structure.

(a) Byte Byte sequence   (b) Record sequence.



(c) Tree.

Three kind g file structure.

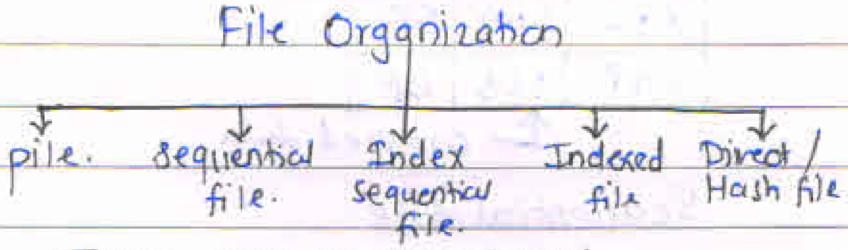
:. In Byte sequence, this unstructure sequence g bytes. OS does not impose any structure on the files. All i/p are seq. g file

- Conside, sequence g file as 8-bit. OS does not interprete the files
- In this there is max^m flexibility for user.
- Drawback is that there is no support from OS side.

- In Record seq, it is fixed sized record. data can be read or written But record can't be inserted or deleted in the middle g file
- In tree, it is tree g disk block. each block hold no key record.
- Record can be searched By key value. & new record can be inserted anywhere in the file structure. Tree is sorted on key field.

* File Access Method : (File Organization)

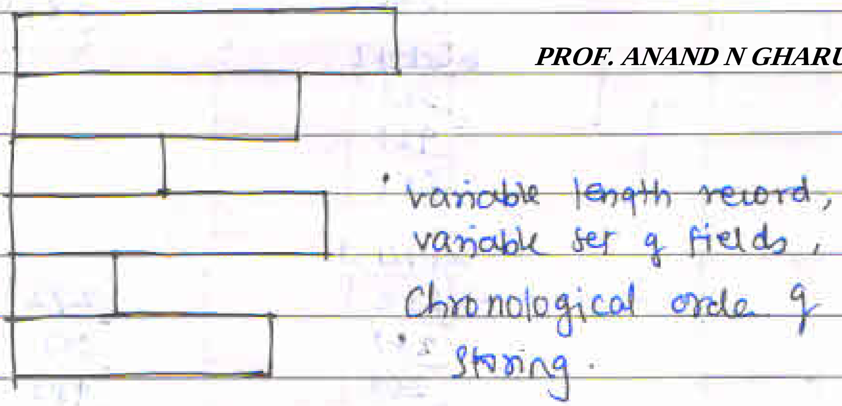
- The file is collection of records where each record consist of one / more fields.
- file org^n can be defined as methods of storing data records in a file.
- The primary objective of file org^n is to provide means for record retrieval and update.
- Some file org^n method :



FILE ORGANIZATION.

1) The Pile :

- It is simplest form of file org. data are stored in order as they arrive.
- A pile is used to accumulate the mass of data & save it.
- In pile, record may have different fields or similar field in diff. order.
- There is no structure of pile file.
- To search a record, it is necessary to search each record until the desired record is found.
- piles files uses less space when stored data vary in size & structure.



PILE FILE.

2) Sequential File :
- fixed format is used for record.
- All record are of same length
- posi^n of each field in record and length of field is fixed.
- Record are physically ordered on the value of one of the fields called the ordering fields.

Name	Roll No	Year
ABC	101	TE
XYZ	102	SE
PQR	103	BE.

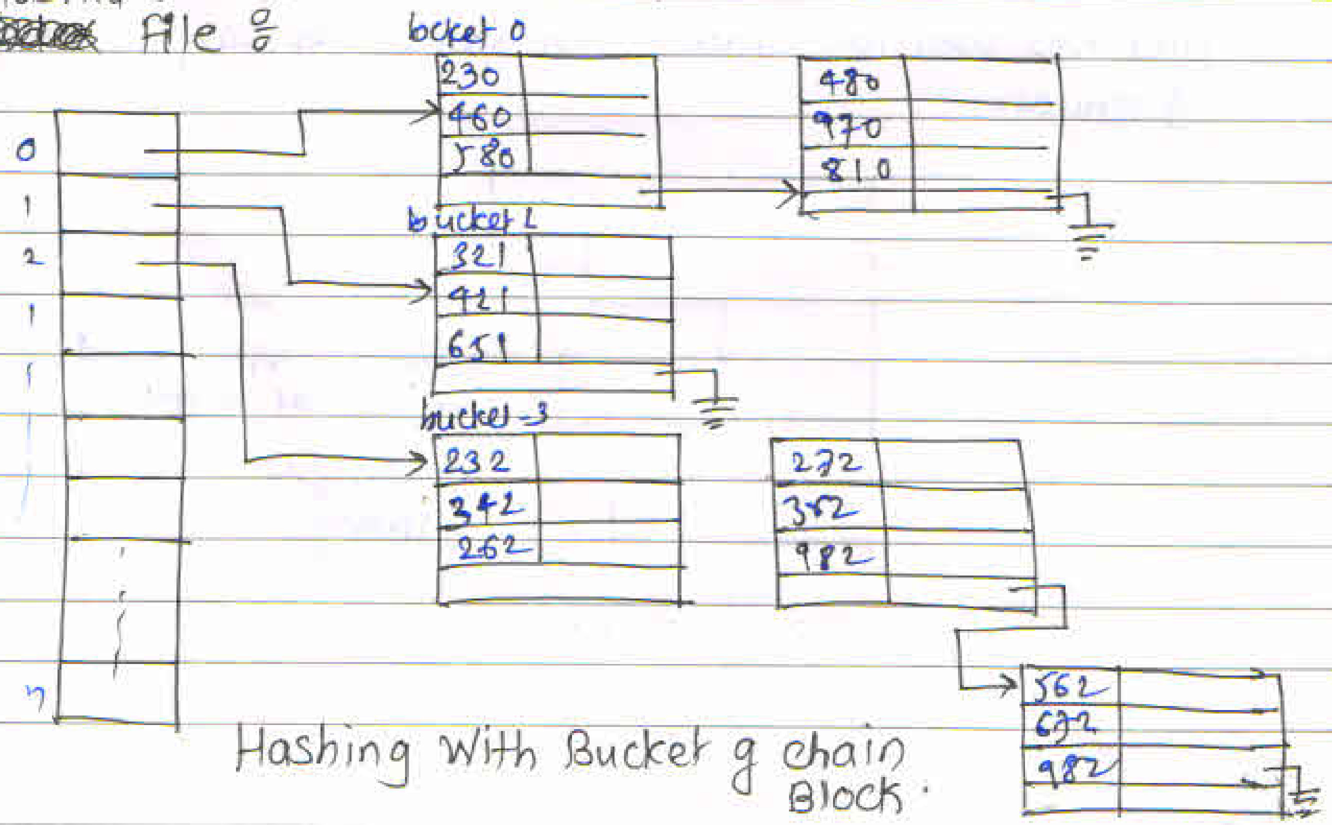
↑ ordered data.

Sequential file.

* Advantages :-
- easy to read data in order
- easy to search data bcz next record is found in same block.
- Searching order is fast. Binary search can be used.

Disadvantages :-
- non-ordering search is not allowed.
- inserting a record is an expensive task
- Deleting a record is too difficult
- modification of field might take more time.

3) Hashed (Direct) file :



Hashing With Bucket of chain Block.

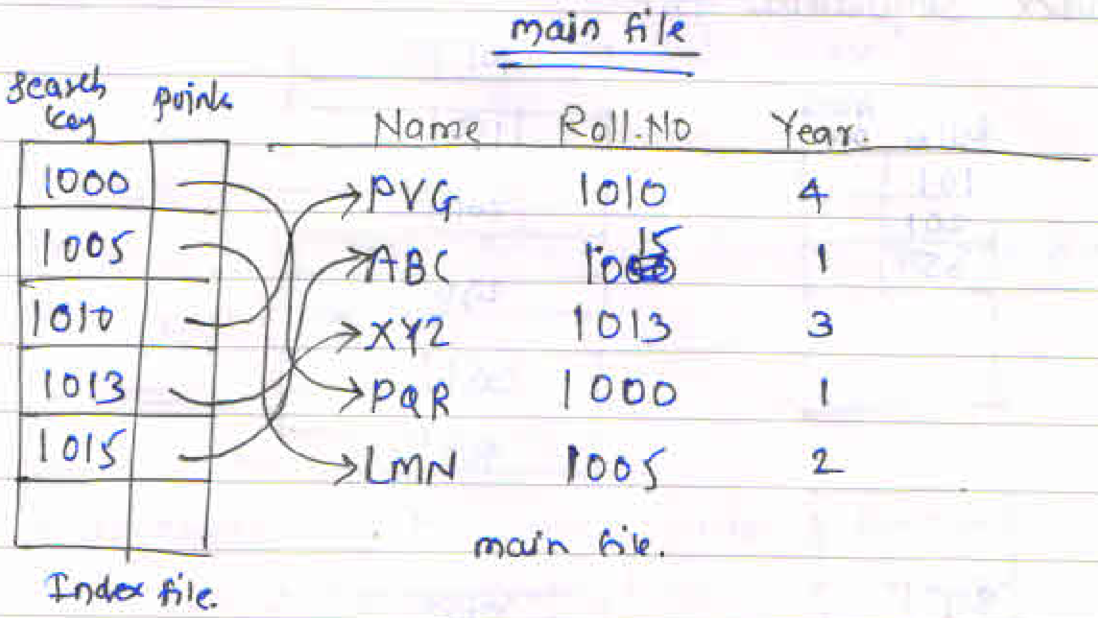
Hashed (Direct) file Organization :-

- It is common tech used for fast accessing of records on secondary storage.
- Record of files are divided among bucket. A bucket is either one disk block or cluster of contiguous block.
- A hashing fn map a key into a bucket number. a bucket are numbered from 0, 1, 2 ... b-1.
- A hash fn f maps each key value into one of integer thro. 0 to n-1.
- If x is key, f(x) is nos of bucket that contains the record with key x.
- The block making up each bucket could either be contiguous block or they can be chain together in linked list.

$$= \text{bucket directory} + \frac{\text{Nos of Record}}{\text{Nos of bucket} \times \text{nos of record per block}}$$

Thus, the opn is b time faster (b = nos of bucket) than unordered file.

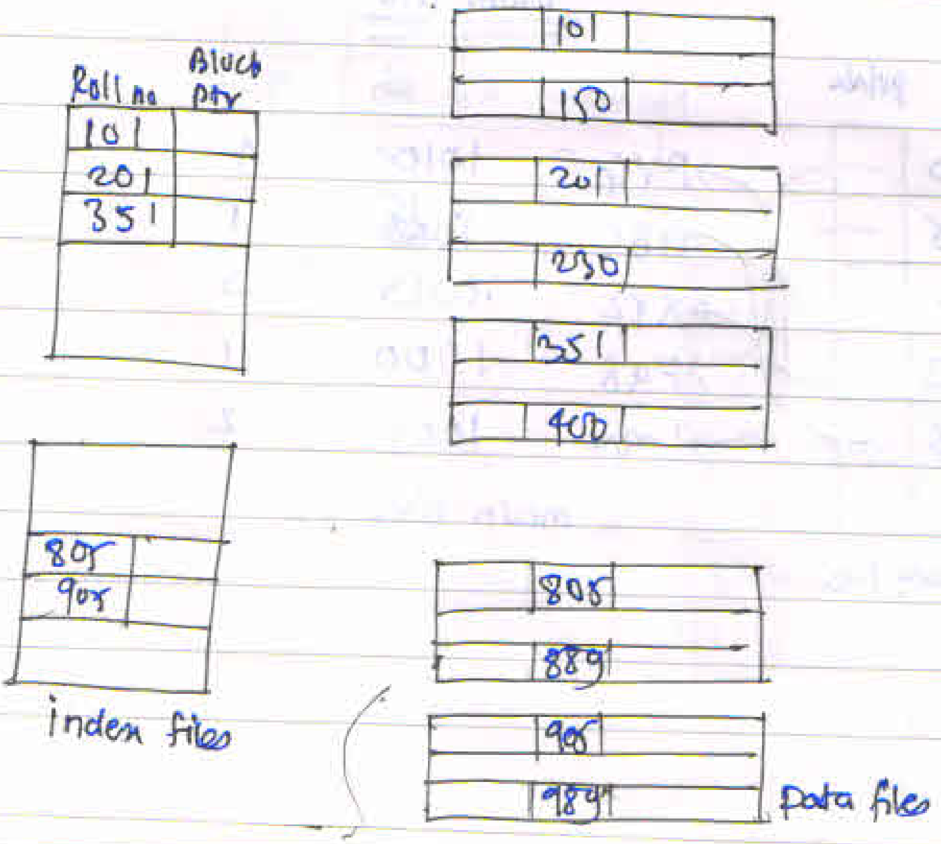
* Indexed File :-



- Indexing is used to speed up retrieval g record. It is done with help g seperate sequential file.
- Each record g inden file consist g two field and point. into the main file.
- To find specific record for given key value, inden is search for given key value.
- Binary search can be used to search in index file. After getting the address g record from index file. the record in main file. can easily be retrieved.
- Index file is ordered on the ordering key i.e. Roll No. each record g index file points to the. corresponding record. main file is not sorted.

Advantages g indexing over sequential file:
- Indexing provides much better flexibility
- It require less storage space.
- with indexing, new record can be added at end g file. It will not require movement g data (record) as in sequential.

4) Index Sequential file:



* Directory structure ÷

- It is data structure for storing detail about files. A directory typically contains a no. of record one per file.
- each record contains 1) file name
  2) file attribute.
  3) Addr. of disk block where data are stored.

Types of Directory structure ÷
1) Flat directory
2) Hierarchical directory.

1) Flat directory - all files are contained in root directory & there is no other subdirectory.

2) Hierarchical Directory ÷
it is organized in tree data structure., This is root directory. It can subdirectories and files.
- In this files can be grouped together in natural way.
- file can be located quickly.

* path Name ÷
There are 2 types 1) Absolute path name
2) Relative path name.

1) Absolute path Name ÷
- it consist of path from root directory to the files
e.g. root\ABC\pqr\a.txt ←
- ~~root~~

2) Relative path name.
- it is always w.r.t. working (current directory).
Process can change working direct.
1) dot (.) refers current directory
dotdot (..) - refers parent directory.

* File System Implementation : OR File management System :

— The main fn of file sys is to manage space on secondary disk., which includes keeping track of both allocated disk block & free disk block.

The main prob. in allocating files are

- effective utilization of disk space.
- Fast accessing of files
- slow disk access time etc.

There are 3 techniques for allocation of disk block :-

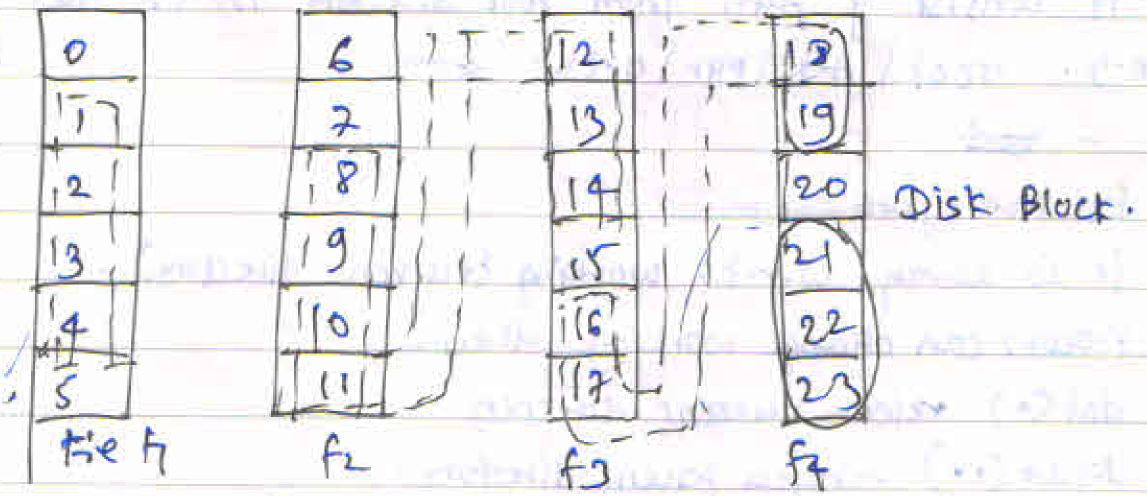
1) Contiguous allocation
2) Linked allocation
3) Indexed allocation

1) Contiguous Allocation :

- In this, files are assigned to contiguous area of secondary storage. as shown in fig.

Directory

File	Start	Size.
f_1	1	4
f_2	8	7
f_3	16	4
f_4	21	3



Contiguous Allocation of files.

* Disk space Management :-

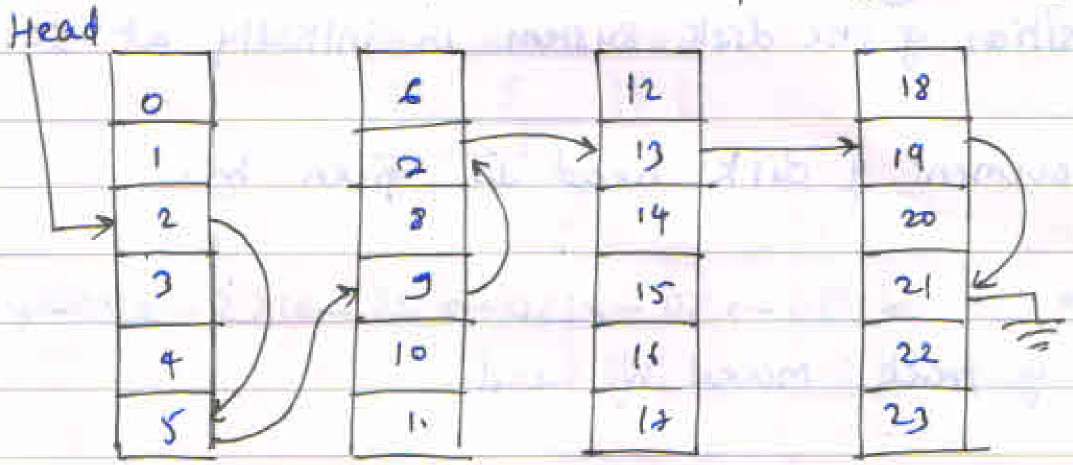
- OS maintain list of free disk blocks.. free disk block are those in a disk which are not being used by any file.
- whenever file is created, the list of free blocks is searched for and allocated to new file.
- The blocks allocated to this file is removed when file is deleted. , it's disk space is added to the list of free block.

There are 2 methode of disk management :-

1) Linked list
2) Bit map.

1) Linked list :-

- In this method, all free disk blocks are linked together by each free block pointing to next free block.
- in e.g., block 2 is the 1st free block of linked list of free block.
- block 2 contain the addr. of free block 5 and so on.
- last free block 21, that is not pointing to any free block.



Linked list free disk Block.

2) Bit map :-

- list of free block is Implemented as a bit map or bit vector
- each block is represented by single bit.
  - 0 for free block
  - 1 for allocated block.

PROF. ANAND N GHARU

e.g. where 2, 5, 7, 9, 13, 19, 21 are free block.
The bit map for free block will be
110110101011101111101 0.

* Disk Schedulling :
- Seek time
- latency
- data transfer rate.

* Disk schedulling Algo :-
1) FCFS schedulling
2) Shortest seek time first schedulling
3) Scan schedulling
4) Circular scan (C-scan) schedulling.

1) FCFS schedulling :
- In this, request are serviced in the seq. as they arrive.
- it is easy to prog.
- It does not perform optimal performance.

e.g.

100, 190, 50, 150, 25, 155, 70 and 85.

we are assuming a disk with 200 track and the head position of the disk system is initially at 50.

- The movement of disk head is given by.

$$50 \rightarrow 100 \rightarrow 190 \rightarrow 50 \rightarrow 150 \rightarrow 25 \rightarrow 155 \rightarrow 70 \rightarrow 85$$

The no of track moved by head.

$$= |50-100| + |100-190| + |190-50| + |50-150| + |150-25| + |25-155| + |155-70| + |70-85|$$

$$= 50 + 90 + 140 + 100 + 125 + 130 + 85 + 15.$$

$$= 735 \text{ tracks}$$

PROF. ANAND N GHARU

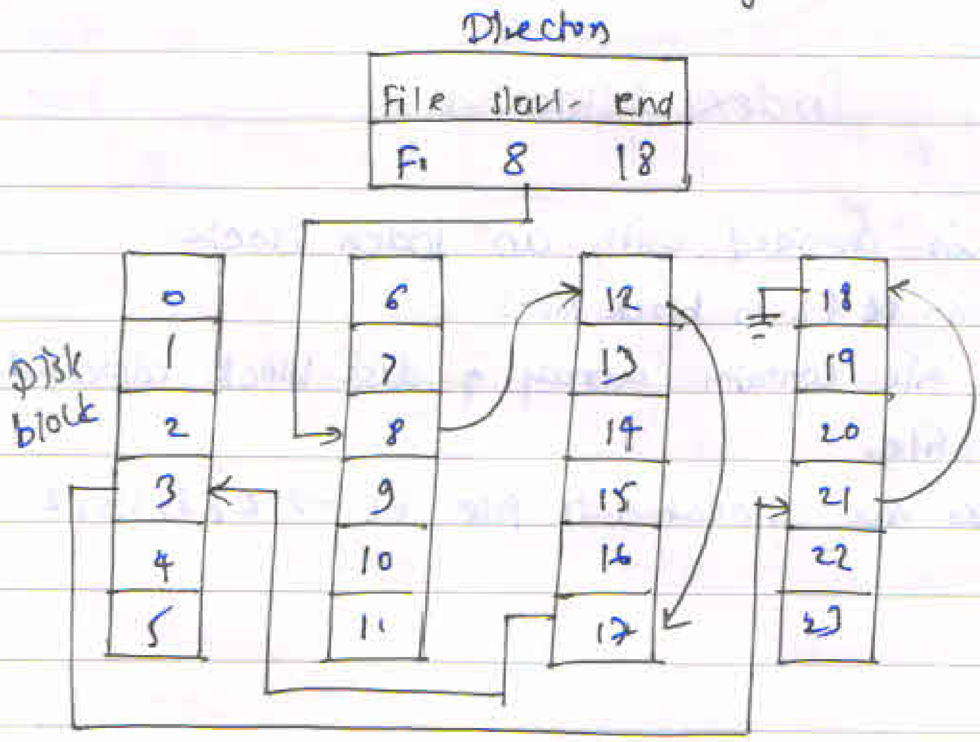
$$\text{Average seek time} = \frac{735}{8} = 91.88 \text{ tracks.}$$

1) Contiguous Alloc^n :-

- In this, user specifies in advanced size of the area needed to hold a file before creation.
- If desire amount of contiguous space is not available then file will not be created.

2) Linked Allocation :-

- Files tends to grow and shrink dynamically. It is very difficult for user to know in advance the size of the file.
- Linked allo^n is essentially a disk based version of linked list.
- In this, alloc^n list of file. to a file is linked list of blocks. These block may be scattered thro. disk.
- These block need not be contiguous.
- Each block contain address of next block.

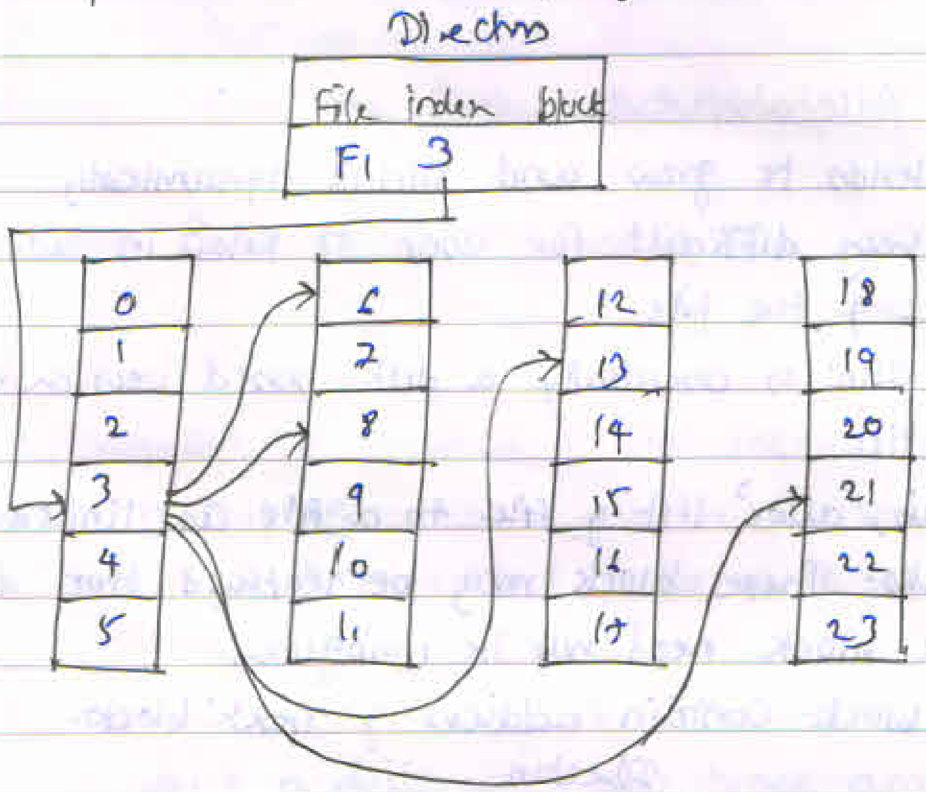


Linked Allocation.

- there is file of 6 blocks which start from block 8. and end with block 18.

3) Index Allocation :-

- one drawback of linked allo is that it does not support direct accessing as allocated block are linked together.
- this problem is overcomed by indexed allocation



Index Allocation.

- each file is provided with an index block. index for file f1 is block No. 3
- An index file contain array of disk block allocated to the said file.
- foll. blocks are allocated to file f1 → 6, 8, 13, 21

* Shortest seek time First (SSTF) schedulling :-

- It select the disk I/O request that requires the least movement of disk head from it's current position.
- Since seek time is generally proportional to the track difference bet^n the request, this approach is implemented by moving the head to the closest track in the request queue.
- It cuts total head movement considerably
- It does not provide fairness (no waiting for long time).

e.g. 100, 190, 50, 150, 25, 155, 70 and 85.

we are assume disk with 200 track and head position of disk system is initially at track 55.

Once we track 55, the next closest track is 50. From here it go to 70 and then 85 and so on.

The movement of head is

Next track accessed	Nos of tracks traversed
50	55-50 = 5
70	70-50 = 20
85	85-70 = 15
100	100-85 = 15
150	150-100 = 50
155	155-150 = 5
190	190-155 = 35
25	190-25 = 165

Total = 310

PROF. ANAND N GHARU

Average seek length $= \frac{310}{8} = 38.75$ tracks.

3) Scan schedulling :-

- it is also called as "elevator" algorithm.
- it is developed for 2 reasons
  1) To procure advantages of both FCFS & SSTF
  2) To eliminate short coming of both FCFS & SSTF algo.
- In this algo, R/W head of disk start from one end and moves towards the other end, servicing all outstanding request enroute, until it reaches the last track in that direction. or until there are no more request in that direction.
- The R/W head direction is then reversed and the scan proceeds in the opposite direction, again serving all request in order.
- e.g. 100, 190, 50, 150, 25, 155, 70 and 85.

we assuming disk with 200 track. Initial posi of track is 55.

Next track accessed	nos of track traversed
70	70-55 = 15
85	85-70 = 15
100	100-85 = 15
150	150-100 = 50
155	155-150 = 5
190	190-155 = 35
50	50-190 = 140
25	25-50 = 25

Total = 300

avg seek length = $\frac{300}{8}$ = 37.5 track.

* Circular scan (C-Scan) Algorithm

— In this, scanning & servicing are restricted to one direction only. Thus when the last track is serviced in one direction, the disk head is returned to opposite end of the disk and then scan begin again.

e.g.

100, 190, 50, 150, 25, 155, 70 & 85.

initial = 55

next track accessed	nos of track traversed
70	70-55 = 15
85	85-70 = 15
100	100-85 = 15
150	150-100 = 50
155	155-150 = 5
190	190-155 = 35
25	190-25 = 165
50	50-25 = 25

Total = 325

Avg seek length = $\frac{325}{8}$ = 40.6 track

- It does not cause starvation
- it is better for sys. place of heavy load on the disk.
- C-Scan perform much better than FCFS with service guarantee.

PROF. ANAND N GHARU

e.g.

A disk drive has 640 cylinder numbered 0-639. The drive is currently serving the request at cylinder 68. The queue of pending request in FIFO order is:

84, 153, 32, 128, 10, 133, 61, 69.

starting from current head posⁿ, what is the total distance that disk arm moves to satisfy all the ~~Good~~ pending request for full disk scheduling

1) SSTF 2) SCAN 3) C-SCAN.

Ans:- 244 = 228 = 279

PROF. ANAND N GHARU