UNIT-4

Introduction to File Management

File :-

→ File is a collection of logical records.

→ File is a inter related records.

→ A file is named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks.

→ In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user.

File Structure :-

A file / structure should be according to a required format that the operating system can understand.

- A file has a certain defined structure according to its type.
- A text file is a sequence of characters, organised into lines.
- A Source file is a sequence of procedures and functions.
- An Object file is a sequence of bytes organised into blocks that are understandable by the machine.
- When Operating System defines different file structure it also contai the code to support these file structure UNIX, MS-DOS support minimu number of file system structure.

File type :-

→ File type refers to the ability of the operating system to distinguish different types of file such as text files source files and binary files etc. Many operating system support many types of files. Operating System like MS-DOS, and unix have the following types of files.

1. Ordinary file :-

→ These are the files that contain user information.

→ These may have text, databases or executable program.

→ The user can apply various operations on such files like add, modify, delete or even remove the entire file.

2. Directory file :-

→ These files contain list of filenames and other information related to these files.

3. Special files :

→ These file are also known as device files

→ These files represent physical device like disks, terminals, printers, networks, tape drivers etc.

→ These files are of two types :

- Character Special files :- Data is handled character by character as in case of terminals (or) a printer.

- Block Special files :- Data is handled in blocks as in the case of disks & types.

Operation On files :

1. Search a file :- Any user wants to searching of any file is available or not available under directory.

2. Create a file :- Any user wants to creating of any file is done with the help of using create () file operation. Create of file is used to Properly create a new file.

3. Delete of file :- Any user wants to deleting of particular file from directory.

4. Transfer a file :- To transfer a file from one directory to another directory.

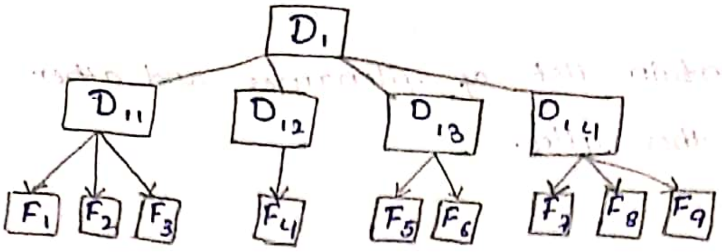
5. list a file / list a directory :- To grouping of no. of files into single at that time, we use list a file.

Directories :-

→ A directory is the collection of the correlated files on the disk.

→ In a directory, we can store the complete file attributes, on Some attributes of the file. A directory can be comprised of Various files.

→ With the help of directory, we can maintain the information related to the files.

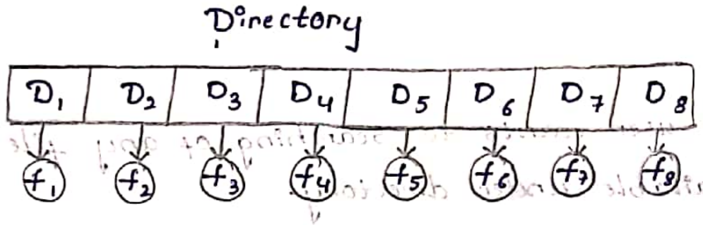


1. Single level directory :

→ Single level directory is the easiest directory structure. There is only one directory in a single-level directory and the directory is called a root directory.

→ In a single level directory, all the files are present in one directory that makes it easy to understand.

→ In this under the root directory, the user cannot create the sub directories.

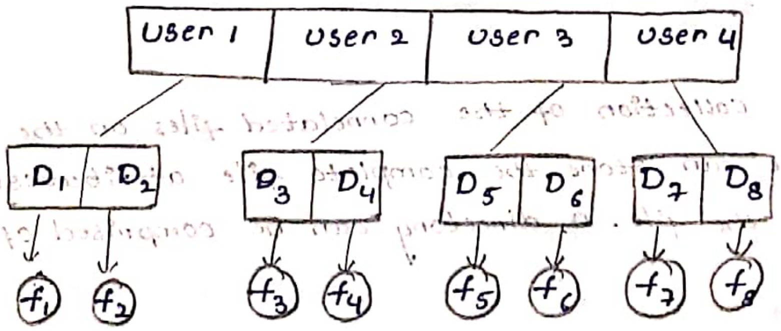


2. Two-Level directory :

→ Two-level directory is another type of directory structore, in this, it is possible to create an individual directory for each of the users.

→ There is one master node in the two-level directory that include an individual directory for every user.

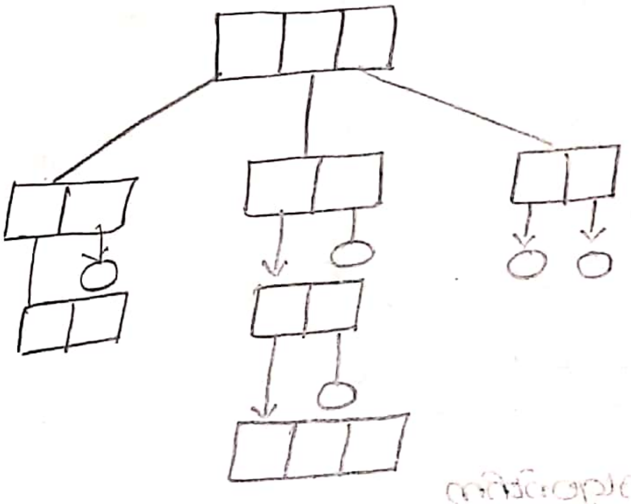
→ At second level of the directory, there is a different directory present for each of the users, without permission, no user can enter into the other users directory.



3. Hirerachical directory :-

→A tree structured directory is another type of directory structure in which the directory entry may be a sub-directory or file.

→The tree structured directory reduces the limitation of two-level directory. we can group the same type of files into one directory.



overview of the DISC structure and attachment :-

→It maintains a spindle

→Spindle is cylindrically type

→Spindle is interconnected with no. of Circular representations.

→Each and every Circular representation is called Floppy disk (or) CD (compitable disk) or DVD (Disk Video Device).

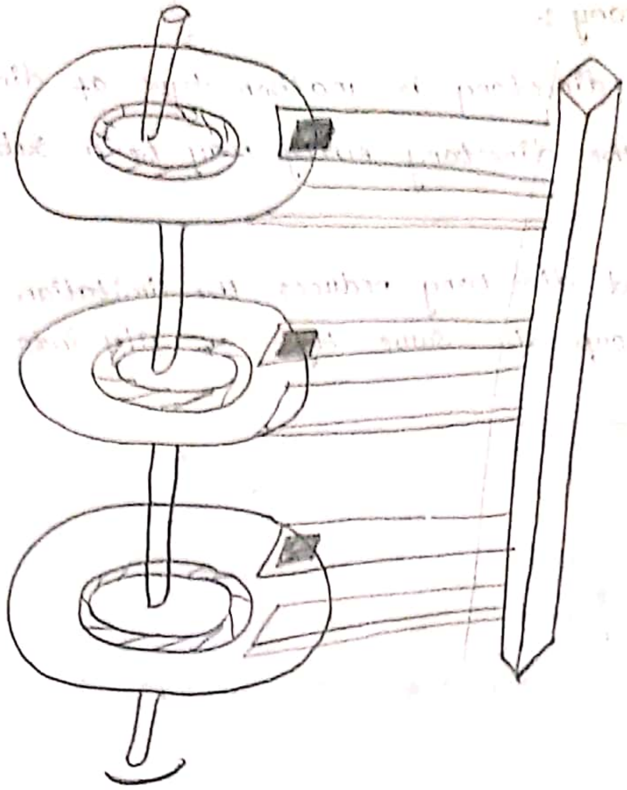
→Each and every CD/DVD has maintained a surface. Surface is filled with magnetic material or magnetic core.

→Available CD/DVD is attached with read and write heads with help of disk / actuator.

→Read head is the process of retriving/reading any information from the CD/DVD.

→Write head is taken care of to properly sending of any information from private area into either CD/DVD.

→CD/DVD are rotated with the help of motor.



* Disc Scheduling Algorithm

* FCFS :- FCFS stands for First come First serve. FCFS disc scheduling algorithm is based on header value and blocks of string

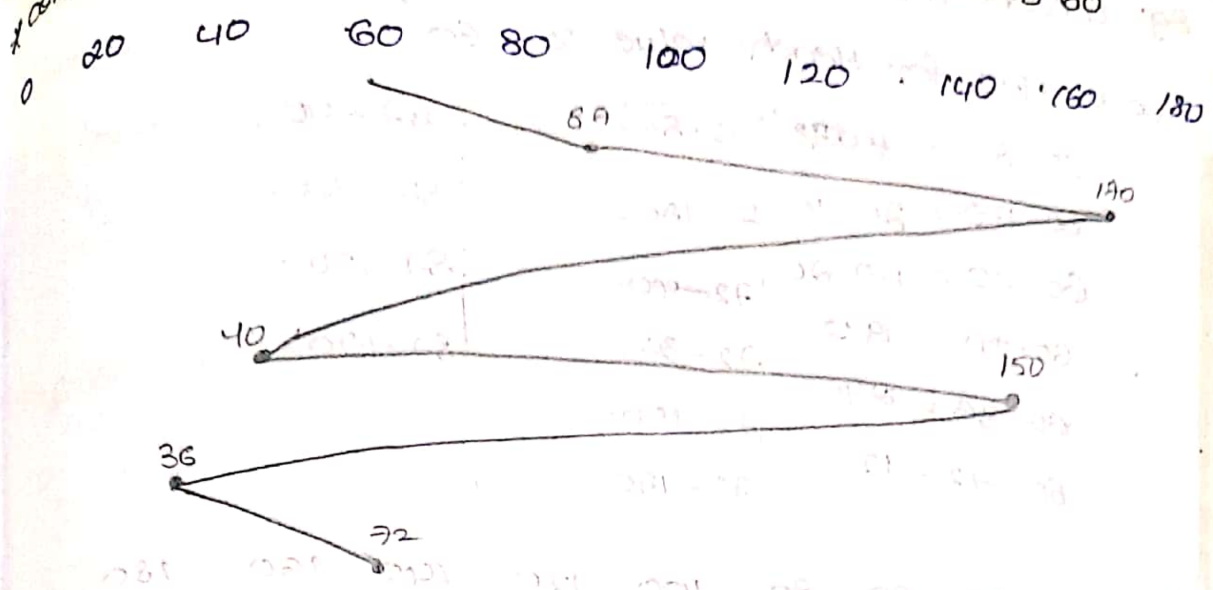
→ According to FCFS scheduling algorithm, available string information required to arrange either ascending order (or) descending order.

→ FCFS algorithm header movement starts with forward direction.

→ Finally calculated no. of cylinders with help of using formula:

$$\frac{\text{Total deviations b/w header value \& block of string}}{\text{Total blocks of string}}$$

* consider a string : 87, 170, 40, 150, 36, 72. Header value is 60



$$\text{No. of cylinder} = \frac{D(60 \text{ to } 87)+D(87 \text{ to } 170)+D(170 \text{ to } 40)+D(40 \text{ to } 150)+D(150 \text{ to } 36)+D(36 \text{ to } 72)}{6}$$

$$= \frac{27+83+130+110+114+36}{6}$$

$$= 83.33 \text{ cylinders}$$

*SSTF :-

→ SSTF stands for shortest seek time first

→ SSTF initially required string (string contains no. of blocks)

→ SSTF implementation is based on seek time.

→ Seek time means to find out difference b/w header value and block of string.

→ According to SSTF, needed to arrange string information either ascending or descending order.

→ Header movement starts with least seek time value.

→ Finally calculated no. of cylinders using math formula.

$$\frac{\text{Total deviations b/w header value \& block of string}}{\text{Total block of strings}}$$

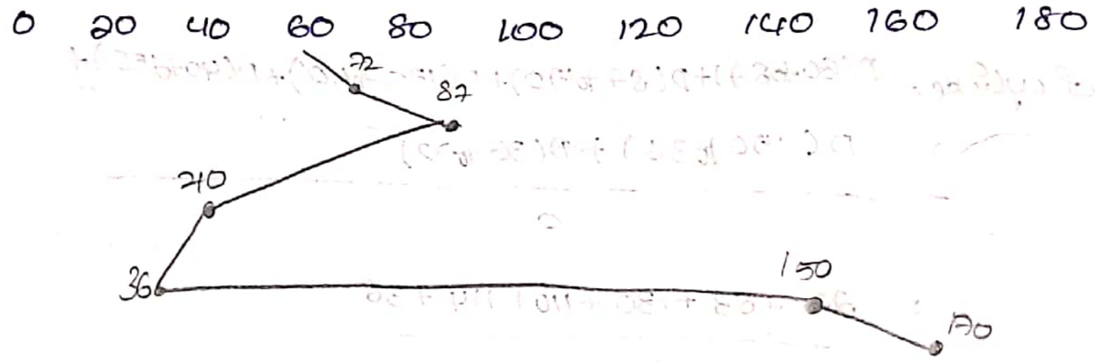
Eg:- consider a string 87, 170, 40, 150, 36, 72. header value is 60

Seek time for Header value is 60

60 - 87 = ~~110~~27
60 - 170 = ~~20~~ 110
60 - 40 = ~~110~~ 20
60 - 150 = 190
60 - 36 = 24
60 - 72 = 12

72 - 87 =
~~72 - 170 =~~
~~72 - 400 =~~
72 - 36 =
72 - 150 =
72 - 170 =

87 - 40 =
87 - 36 -
87 - 150 =
87 - 170 =



$$\text{No. of cylinder} = \frac{D(60\ to\ 72) + D(72\ to\ 87) + D(87\ to\ 40) + D(90\ to\ 36) + D(36\ to\ 150) + D(150\ to\ 170)}{6}$$

$$= \frac{12 + 15 + 47 + 4 + 114 + 20}{6}$$

= 35.33 cylinders

*Scan Algorithm

→ SCAN algorithm initially require string value (string contains no. of blocks)

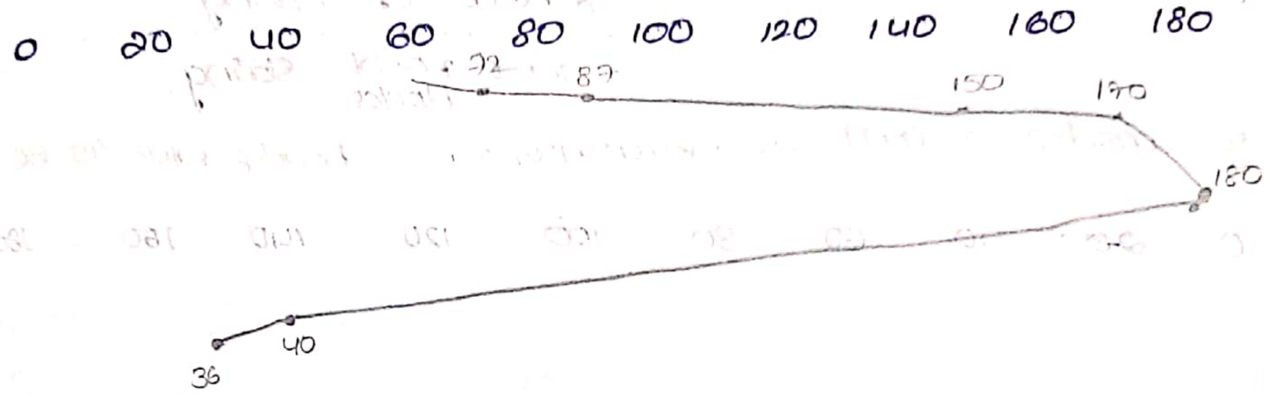
→ According to SCAN algorithm required to arrange available information into either ascending or discending order

→ SCAN Algorithm is implemented based on forward end / backward end / header value

→ finally calculated no. of cylinders using mathematical formula:-

$$\frac{\text{Total deviations b/w header value \& blocks of string}}{\text{Total blocks of string.}}$$

* consider a string 87, 170, 40, 150, 36, 72 header value is 60



$$\text{No. of cylinder} = \frac{D(60 \text{ to } 72) + D(72 \text{ to } 87) + D(87 \text{ to } \overset{150}{40}) + D(150 \text{ to } 170) + D(170 \text{ to } 180) + D(180 \text{ to } 40) + D(40 \text{ to } 36)}{6}$$

$$= \frac{12 + 15 + 63 + 20 + 10 + 140 + 4}{6}$$

$$= \frac{264}{6} = 44 \text{ cylinders}$$

* CSCAN Algorithm

→ CSCAN stands for circular scan

→ It is a disc scheduling algorithm

→ According to cscan algorithm Arranging of given string blocks into either ascending order (or) descending order

→ Csan algorithm header value needed to move forward direction.

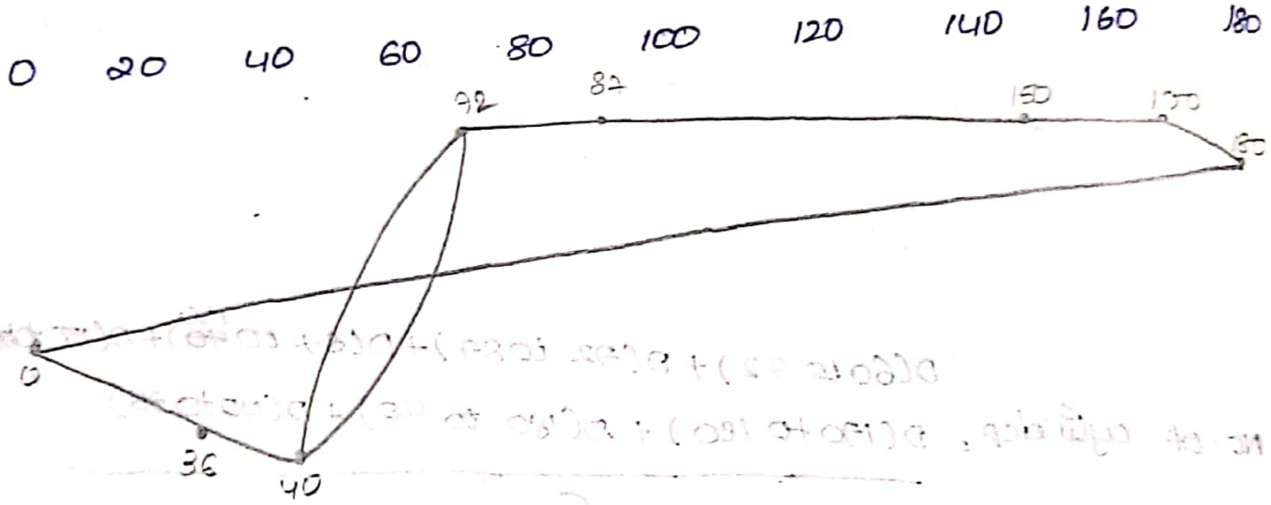
→ According to cscan algorithm header value needed to send from header value to right deadend value and cover all intermediate blocks.

→ According to cscan algorithm needed to send form right deadend value to left side dead end value and covers all intermediate blocks.

→ finally to calculate total no.of cylinders using mathematical formula to

$$\text{Total no.of cylinders} = \frac{\text{Total deviations b/w header value \& blocks of string}}{\text{total no. of blocks string.}}$$

Eg: consider a string. 87, 170, 140, 150, 36, 72 header value is 60



$$\text{No.of cylinder} = \frac{D(60\ to\ 72) + D(72\ to\ 87) + D(87\ to\ 150) + D(150\ to\ 170) + D(170\ to\ 180) + D(180\ to\ 0) + D(0\ to\ 36) + D(36\ to\ 40)}{6}$$

$$= \frac{12+15+63+20+10+180+36+4}{6}$$

$$= \frac{340}{6} = 56.66$$

* LOOK Algorithm

→ It is disc scheduling Algorithm

→ It is used to overcome drawback of the Scan Algorithm.

→ According to look algorithm needed to arranging of given string value ascending $\overset{\text{order}}{\text{value}}$ or descending $\overset{\text{order}}{\text{value}}$.

→ Heads value moves from left to right deadend value of moves from right to left deadend value.

→ (or) moves from (maximum $\overset{\text{\& minimum}}{}$ block value of given string

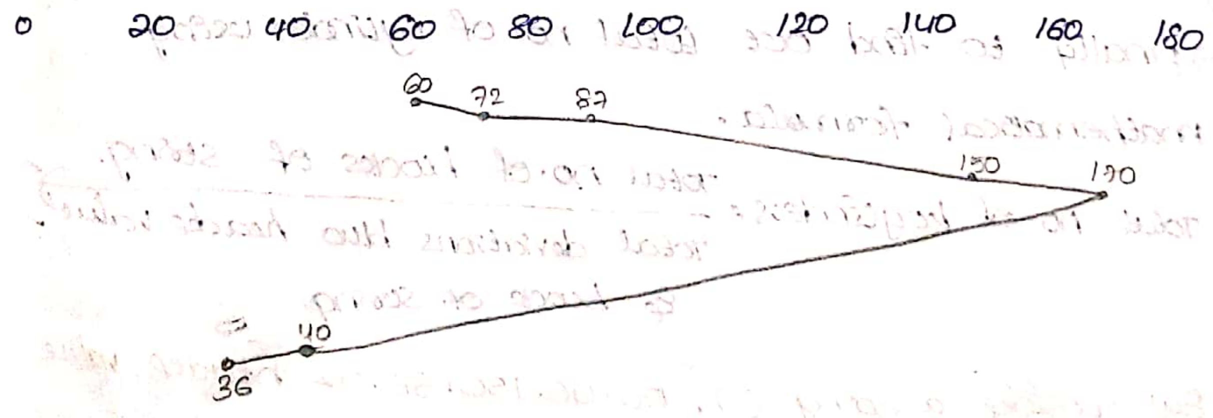
→ According to look algorithm needed to send from maximum. ~~fro~~ highest value to minimum left value.

→ According to look algorithm needed to send from header value to left least lower value and recover all intermediate blocks then needed to send from least highest value to highest least value & so on highest higher value

→ Finally to find out total no. of cyclinders using mathematical formula.

$$\text{total no. of cylinders} = \frac{\text{Total deviations b/w header value and blocks of string}}{\text{Total blocks of string.}}$$

Eg: consider a string 87, 170, 40, 150, 36, 72 header value is 60.



$$\text{No. of cylinders} = \frac{D(60 \text{ to } 72) + D(72 \text{ to } 87) + D(87 \text{ to } 150) + D(150 \text{ to } 170) + D(170 \text{ to } 40) + D(40 \text{ to } 36)}{6}$$

$$= \frac{12 + 15 + 63 + 20 + 130 + 4}{6}$$

$$= \frac{244}{6} = 40.66 \text{ cylinders.}$$

* CLOOK Algorithm

→ It is a disc schedduling algorithm

→ It is used to overcome drawbacks of cscan algorithm

→ According to clook algorithm needed to arranging of given string value into either ascending order or pescending order.

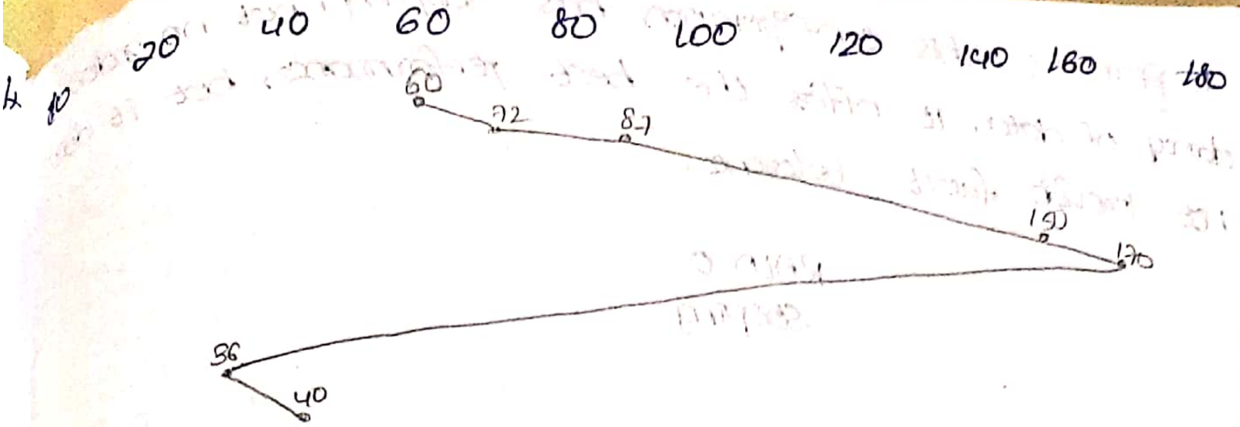
→ Header value moves from left to right maximum highest deadend value & cover all the intermediate blocks

→ According to clook algorithm needed to send from right max highest value to left minimum lowest value then its moves from & so on least highest & it forms a circle b/w header value & left minimum highest value.

→ finally to find out total no. of cylinders using mathematical formula.

$$\text{Total no. of bcylinders} = \frac{\text{Total no. of blocks of string.}}{\text{Total deviations b/w header value \& block of string.}}$$

Eg:- consider a string. 87, 170, 40, 150, 36, 72 header value is 60



$$\text{No. of cylinders} = \frac{D(60 \text{ to } 72) + D(72 \text{ to } 87) + D(87 \text{ to } 150) + D(150 \text{ to } 170) + D(170 + 36) + D(36 \text{ to } 40)}{6}$$

$$= \frac{12 + 15 + 63 + 20 + 134 + 4}{6}$$

$$= \frac{248}{6} = 41.33 \text{ cylinders}$$

* RAID structure

→ RAID or "Redundant Arrays or Independent Disks" is a technique which makes use of a combination of multiple disks instead of using a single disk for increased performance, data redundancy or both.

RAID levels :

RAID 0

RAID 1

RAID 2

RAID 3

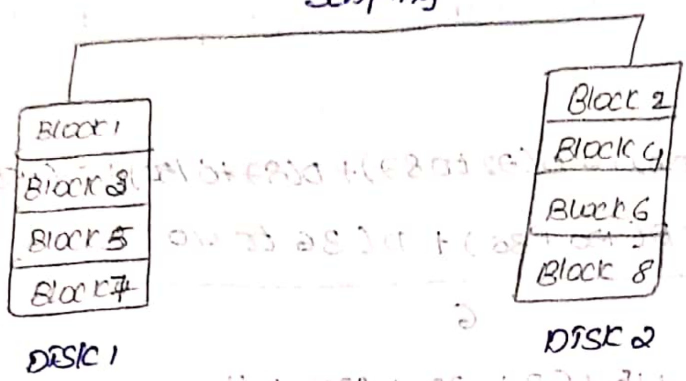
RAID 4

RAID 5

RAID 6

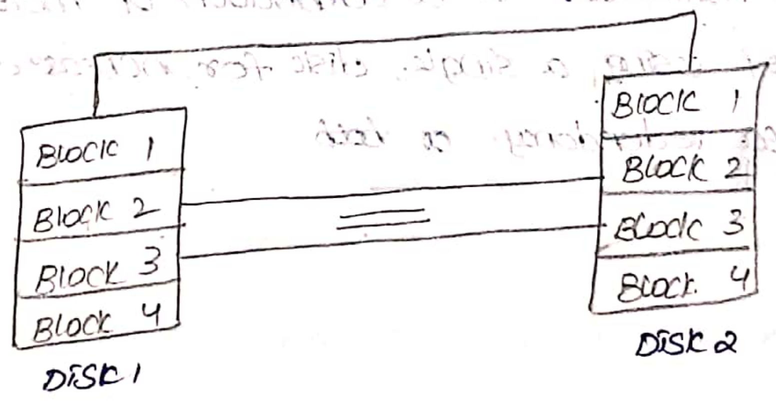
* RAID 0 :- This configuration has striping, but no redundancy of data. It offers the best performance, but it does not provide fault tolerance.

RAID 0
Striping



* RAID 1 :- This configuration also known as disk mirroring. this configuration consists of at least two drives that duplicate the storage of data. there is no striping. Read performance is improved since either disk can be read at the same time. write performance is the same as for single disk storage.

RAID 1
~~Stri~~ Mirroring



RAID 2 :- This configuration uses striping across disks, with some disk storing error checking and correcting (ECC) information. RAID 2 also uses a dedicated Hamming code parity; a linear form of error correction code. RAID 2 has no advantage over RAID3 and is no longer used

RAID 4

BlOCK A1	BlOCK A2	BlOCK A3	BlOCK-A-Parity
BlOCK B1	BlOCK B2	BlOCK B3	BlOCK-B-Parity
BlOCK C1	BlOCK C2	BlOCK C3	BlOCK-C-Parity
BlOCK D1	BlOCK D2	BlOCK D3	BlOCK-D-Parity
DISK 0	DISK 1	DISK 2	DISK 3

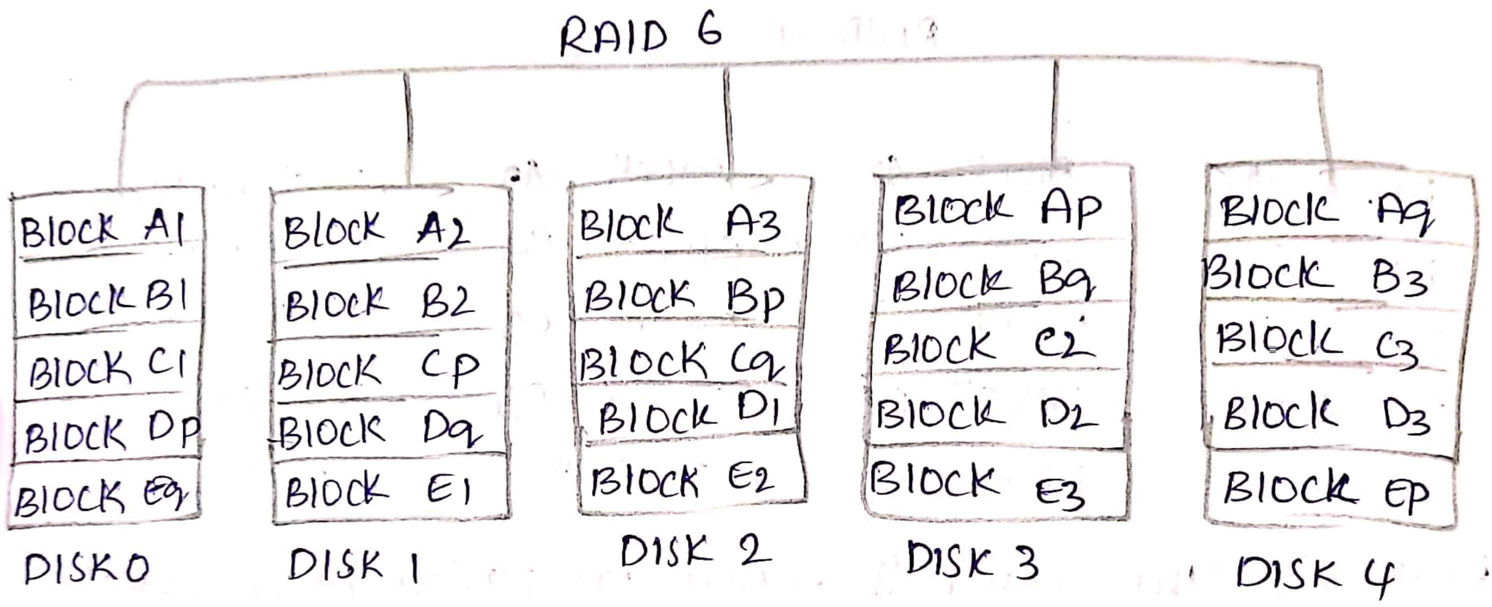
RAID-5 : This level is based on parity block-level striping. The parity information is striped across each drive, enabling the array to function even if one drive were to fail. The array's architechture allows read and write operations to span multiple drives -- resulting in performance better than that of a single drive, but not as high as that of a RAID 0 array. RAID 5 requires at least three disks, but it is often recommended to use at least five disks for performance reasons.

RAID 5

BlOCK A1	BlOCK A2	BlOCK A3	BlOCK-A-parity
BlOCK B1	BlOCK B2	BlOCK-B-Parity	BlOCK B3
BlOCK C1	BlOCK-C-Parity	BlOCK C2	BlOCK C3
BlOCK D0	BlOCK D1	BlOCK D2	BlOCK D3
DISK 0	DISK 1	DISK 2	DISK 3

RAID-6 : This technique is similar to RAID 5, but it includes a second parity scheme distributed accross the drives in the array. The use of additional parity enables the array to continue to function even if two disks fail simultaneously. However, this extra protection comes at a cost. RAID 6 arrays often have slower write performance than RAID 5 arrays.

RAID 6



BLOCK A1	BLOCK A2	BLOCK A3	BLOCK Ap	BLOCK Aq
BLOCK B1	BLOCK B2	BLOCK Bp	BLOCK Bq	BLOCK B3
BLOCK C1	BLOCK Cp	BLOCK Cq	BLOCK C2	BLOCK C3
BLOCK Dp	BLOCK Dq	BLOCK D1	BLOCK D2	BLOCK D3
BLOCK Eq	BLOCK E1	BLOCK E2	BLOCK E3	BLOCK Ep
DISK0	DISK 1	DISK 2	DISK 3	DISK 4