

Unit-5 (I/O management)

PAGE NO.:

DATE: / /

Q1. What are the reasons for not connecting I/O devices directly to system bus? what are the different I/O techniques? Diff. programmed I/O & interrupt I/O techniques.

The reasons for not connecting I/O devices directly to system bus are:-

i) Data Transfer Rate

I/O devices have slower data transfer rates compared to the CPU, which can lead to inefficiencies if directly connected.

ii) Data format & word lengths

I/O devices often have different data format & word lengths than CPU requiring intermediary module like I/O module.

Different I/O techniques:-

i) Programmed I/O (PIO)

In programmed I/O, the CPU directly controls data transfer between peripheral device & memory. It continuously checks the status of the I/O device to determine when data transfer is complete.

2) Interrupt driven I/O

With this technique, the CPU initiates an I/O operation & continues with other tasks. When the peripheral device completes the operation, it sends an interrupt signal to the CPU which temporarily suspends its current task to handle the interrupt & process data transfer.

B. The difference between Programmed I/O & Interrupt-driven I/O techniques lies in how they handle the control of data transfer:-

Feature	Programmed I/O	Interrupt driven I/O
CPU involvement	CPU actively controls data transfer	CPU initiates operation & handle interrupts.
CPU utilization	High CPU utilization during data transfer.	Low CPU utilization during data transfer.
Efficiency	Less efficient due to constant polling	more efficient as CPU performs other task concurrently.
Device status checking	continuous polling of devices status by CPU	CPU is interrupted when device has data transfer.

Redundant Array of Independent Disk (RAID)

One of the important disk organization method is RAID.

RAID (redundant array of independent disks) originally redundant array of inexpensive disks is a storage technology that combine multiple disk drive components into a logical unit.

RAID level 0:-

Create one large virtual disk from a number of smaller disks.

It implements striped disk array where data is broken down into blocks & each block is written to separate disk drive.

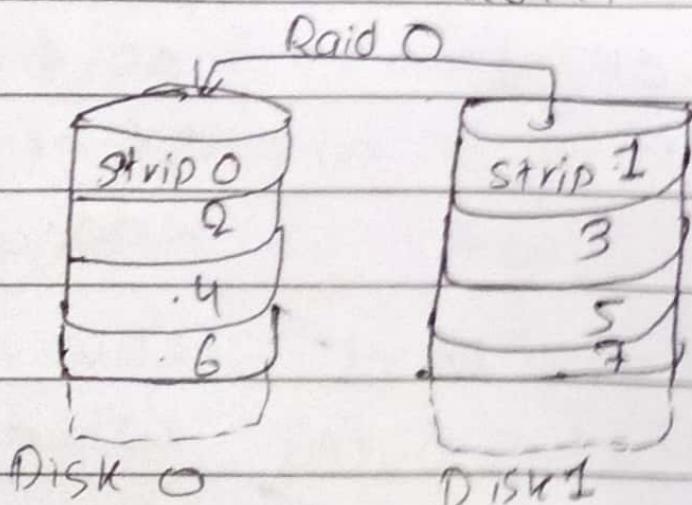
RAID 0 has:-

No redundancy

Improved performance

Additional storage

but no fault tolerance



a) RAID 0 (non redundant)

RAID level 1 :-

It provide disk mirroring.

It is a technique in which data is written to duplicate disk simultaneously.

If one fails instantly switch to other, without loss.



b) Raid 1 (Mirrored).

Raid level 2

It make use of parallel access technique in which all disks participate in execution of I/O request.

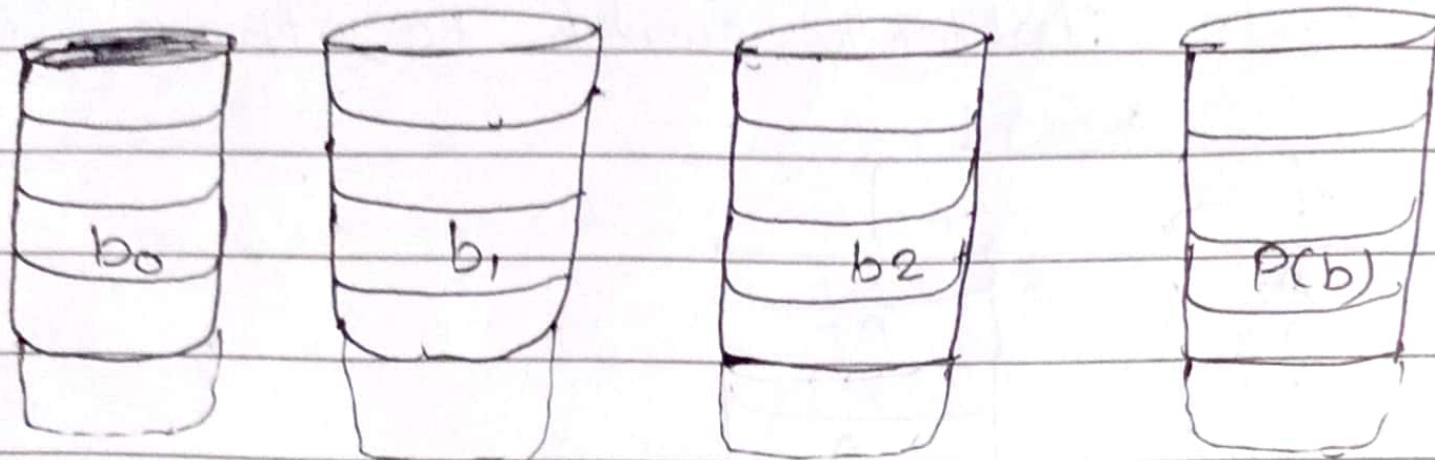
It is effective to use in environment where probability of disk error is high.



c) Raid2 (Redundancy through hamming code).

RAID level 3:

It is similar to RAID 2, difference is that RAID 3 requires only a single redundant disk, ~~and~~ individual bits in same position.



Raid level 4 & 5,

- It make use of independent access technology.

here, a bit - by - bit parity is calculated across corresponding strips on each data disk.

& parity bits are stored in corresponding strip on Parity disk.

(5).

similar to 4 but parity strips are distributed.

Virus & worm. ACL vs ACM.

~~Real-Time Operating System~~
attributes of file

- Q. How file naming is done? Discuss different file allocation methods
- Q. What is file system implementation? Explain link list & i-node file system implementation.
- Q. Describe Access control matrix & Access control list using an appropriate example. How it achieves a level of security in files?
- Q. How files can be allocated using linked list & i-node method? Describe using appropriate figures.
- Q. Define relative and absolute path. How file is implemented in a disk using contiguous, linked list and indexed allocation strategy? Explain their merits & demerits.
- Q. What is ACL? Describe file system implementation methods.
- Q. How files & directories are stored in memory such that they can be managed efficiently? Explain approaches.
- Q. File system allocation technique, what is file?
- Q. What is difference b/w absolute and relative path name of a file? What criteria should be used to decide which strategy? (contiguous, linked, indexed, allocation) is best utilized for a particular file.
- Q. Give a scenario where choosing a large file-system block size might be a benefit, given an example where it might be hindrance.

Q What is file?

File is a collection of data or program stored on a storage medium, identified by a unique name and organized within a file system.

There are different types of files such as data files, text files, program files, directory files & so on.

A file system is used to control how data is stored and retrieved.

Q. How file naming is done? Discuss different file allocation methods.

⇒ File naming is the process of assigning a unique string to identify a file stored on a computer's file system.

Different operating systems have varying rules and restrictions for file naming. Basic rule for file naming include:-

- 1) Using alphanumeric characters (letters & digits).
- 2) Avoiding punctuation marks, accented letters, and non-latin characters.
- 3) Limiting the length of the filename (e.g. 8 characters in DOS or 256 characters in GUI mode OS).
- 4) Ending filenames with a three letter file extension indicating the file type (e.g.: .doc, .xls, .jpg).

File system allocation techniques are methods used by operating system to allocate storage space for files on disk drives efficiently. There are several common file system allocation techniques, each with its own advantages & disadvantages.

Here are some of the most prevalent ones:-

Contiguous Allocation:-

In contiguous allocation, each file occupies a contiguous set of blocks on the disk.

Eg: On 1-kb block disk, a 50-kB file occupies 50 consecutive blocks.

Supports both sequential & direct access methods.

Advantages:-

Simple to implement:

- Requires tracking two numbers per file: disk address of first block and no. of blocks.

Excellent read performance

Thus, contiguous allocation is simple to implement with high performance

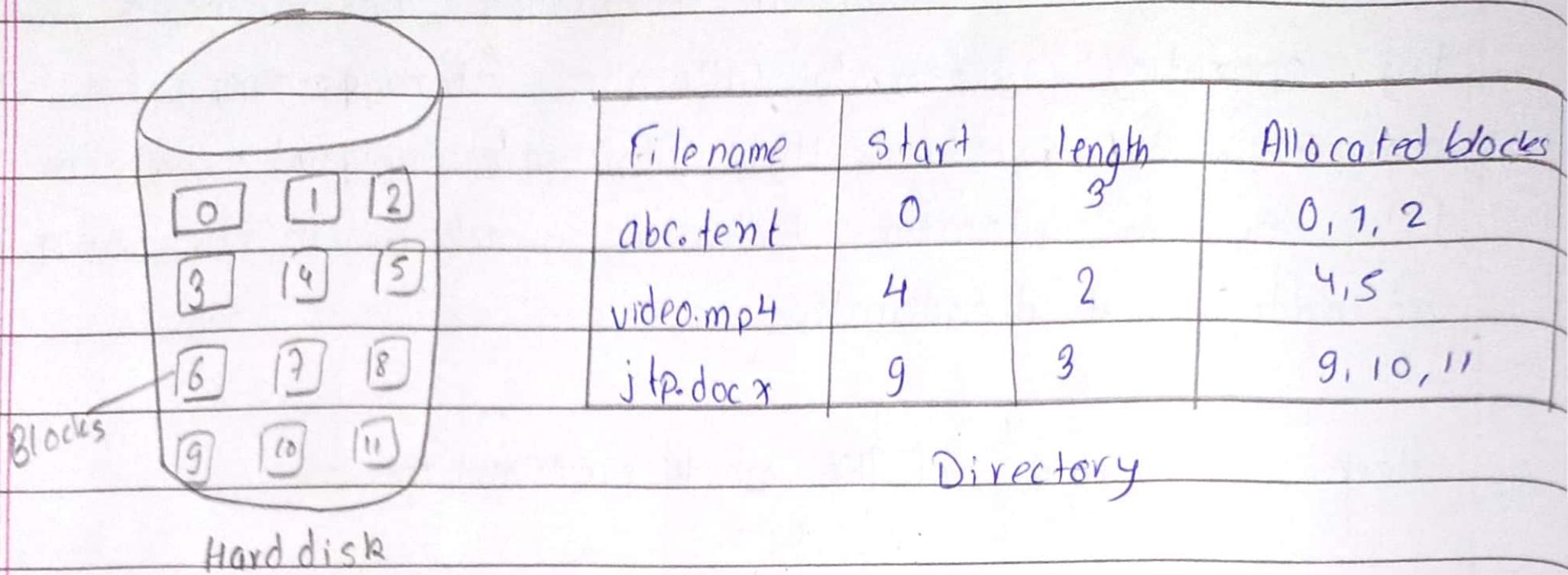
Disadvantages:-

Fragmentation over time

- Disk becomes fragmented with files & holes.
- It needs compaction to avoid this.

Example:-

CD & DVD ROMs.



Contiguous Allocation

Linked list allocation

This method keeps each file as a linked list of disk blocks as shown in the following figure. The first word of each block is used as a pointer to the next one. The rest of block is for data.

Advantage:

No external fragmentation.

No need to declare the size of the file.

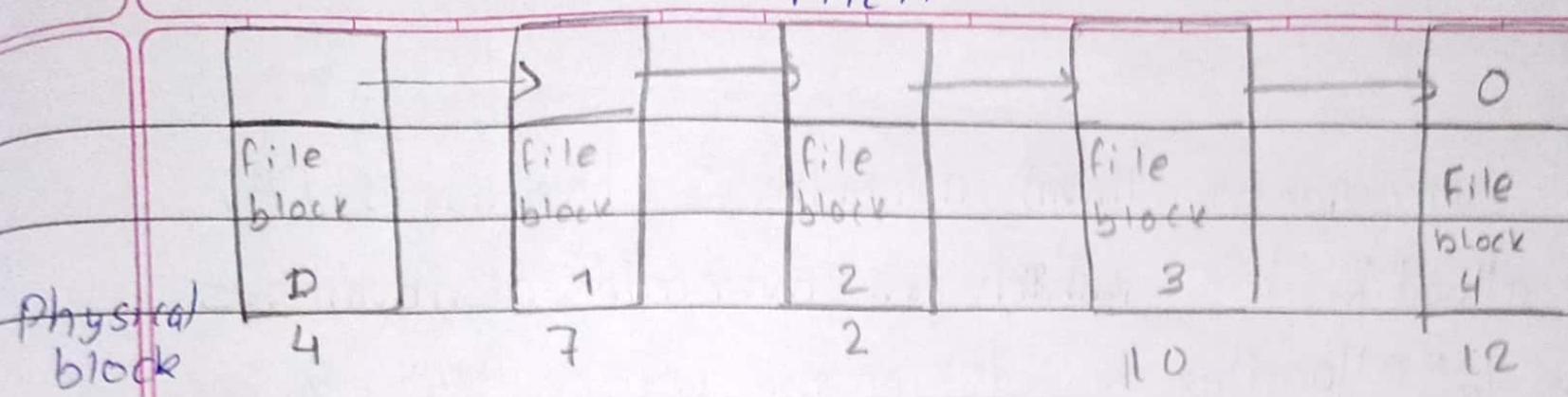
It requires remembering the disk address of first block only.

Disadvantages

Inefficient to support direct-access. (effective only for sequential access files).
Another severe problem is reliability. (pointer can lost).

Few amount of block space is occupied by the pointer.

File A



File B

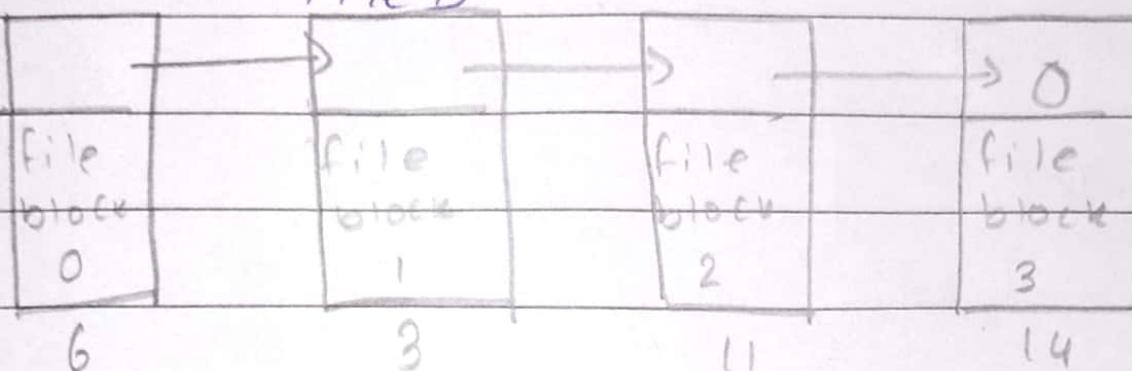


Fig: Modeling a file as a linked list of disk blocks.

3. FAT (file allocation table)

Physical block	
0	
1	
2	10
3	11
4	7 ← file A starts here
5	
6	3 ← file B starts here
7	2
8	
9	
10	12
11	14
12	-1 ← file A ends here
13	
14	-1 ← file B ends here
15	← Unused block

Fig 1 Linked list allocation using a FAT in main memory Gurukul

An imp variation on linked allocation is the use of FAT i.e. file allocation table which can overcome disadvantages of linked-list allocation by taking the pointer word from each disk block & putting it in a table in memory called file allocation tabb.

Used by MS-DOS.

Advantages:-

Entire block is available for data.

Random access is much easier.

Disadvantages:-

Entire table must be in memory all the time for work

4. Indenred allocation.

The indenred allocation method is the solution to the problem of both contiguous and linked allocation.

Advantages.

- Supports direct access to file blocks without enternal fragmentation.
- Only requires remembering the disk address of the first block, subsequent blocks can be located starting from there.

Inden allocation solves the problem by bringing all pointers from each disk block & putting together into one location called inden block.

Disadvantages:-

Index table must be kept in memory at all time.

Index block incurs space overhead.

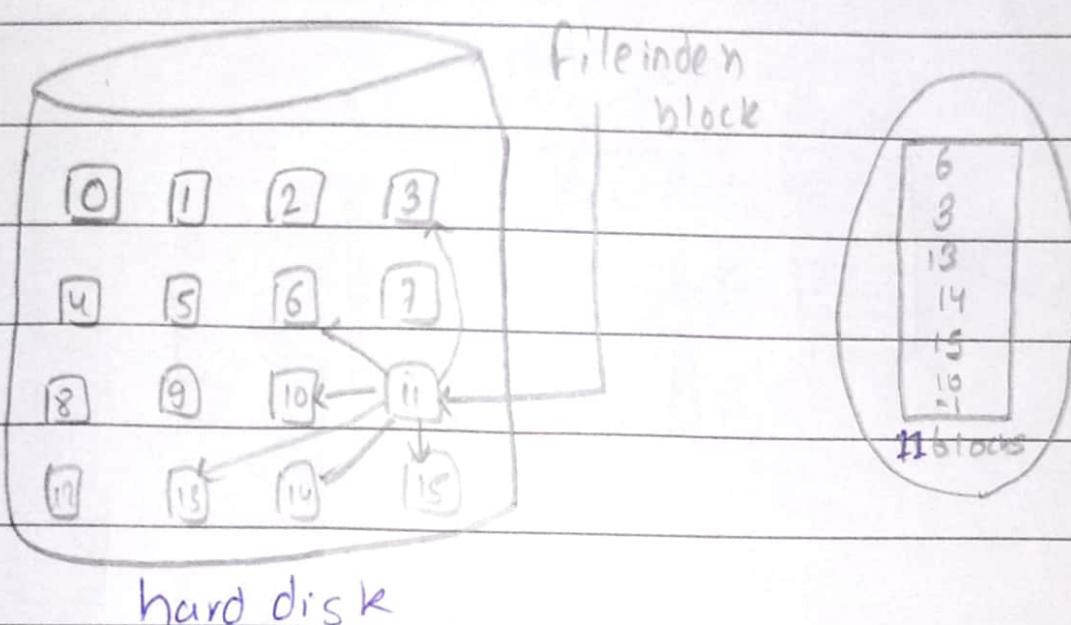


Fig: Index allocation

5. I-node

One of the most modern method for keeping track of which blocks belongs to which file is to associate with each file a little table called I-Node (Index-Node).

File Attributes

Address of disk block 0	→	
" " " " 1	→	
" " " " 2	→	
" " " " 3	→	
" " " " 4	→	
" " " " 5	→	
" " " " 6	→	
" " " " 7	→	
n of block of Pointers		Disk block containing additional disk address

I-node

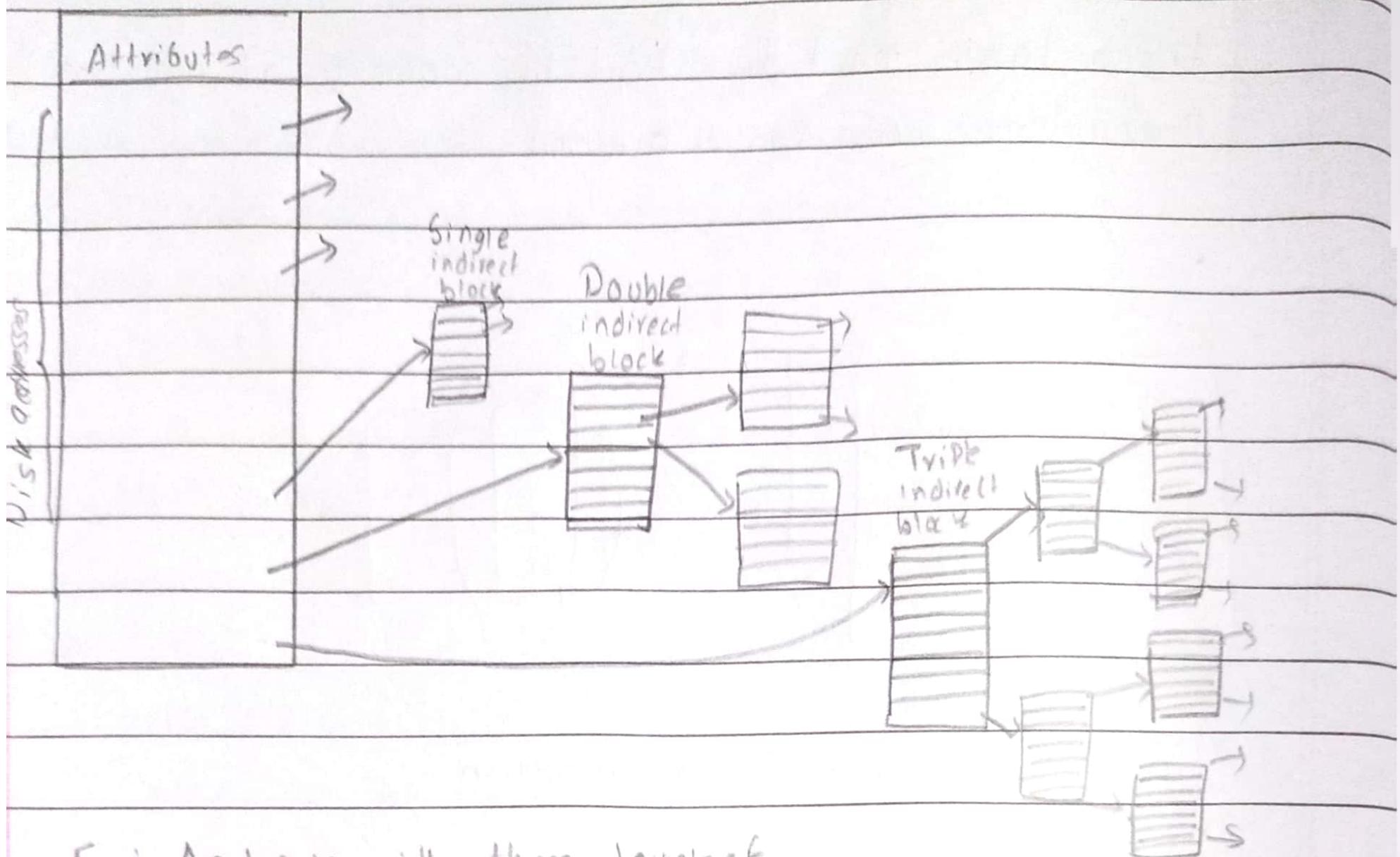


Fig: An I-node with three levels of
indirect blocks

First few disk address in I-node. (small files)

Somewhat larger - single indirect block.

If still not enough - double " "

If also not enough - triple " ".

What is file system implementation?

File system implementation is the development & integration of software components that organize & manage files & data within an OS, encompassing design, data structures, allocation methods, etc.

Define relative and absolute path.

→ Path is the route through a file system to a particular file. Pathname is the specification of that path.

Two common methods:

Absolute pathname.

Absolute pathname starts from the root directory & specifies all directories leading to the desired file or directory.

Example: D:\NAST\Bachelor\BE compute\OS.

indicates a file named OS within

BE computer directory, - subdir.

which is within Bachelor directory, - subdir.

which is within NAST directory, - subdir.

all located within root directory - (D)

Relative path name

Relative path name specifies locations relative to the current working directory.

- Relative pathname is a shortcut navigating up or down in a directory structure to locate a file or folder.

It uses "... " to mean "go up one level".

Relative path do not begin with "\".

Eg:- "... \NAST\Bachelor\BE compute\OS."

means, "go up one level, then find folder NAST, inside it Bachelor, then BE computer & finally OS.

Aspect	Absolute Path name	Relative Path name.
Definition	Specifies the full directory path starting from root directory	Specifies the path relative to the current working directory
Starting point	Always start from root directory	Start from current working directory
Use of symbols	may include drive letter (e.g. C:\) & \ or root directory	does not include root directory symbol; uses "\---" to move up & directory names to down.
Portability	less portable	More portable; easier to move files.
e.g:	C:\Users\Username\Documents\file.txt	\---\Documents\file.txt

How files & directories are stored in memory such that they can be managed efficiently? Explain approaches.

Explain

FAT, linked list, Indirect,

⇒ Files & directories are stored in memory using various approaches to ensure efficient management. Here are some common approaches explained:-

1. Contiguous Allocation

- - - - -

2. linked list Allocation

- - -

3. Indirect Allocation

- - -

4. FAT

- - -

By employing these storage approaches file system can efficiently manage files & directories in memory, balancing factors like performance, fragmentation & ease of implementation.

Give a scenario - - -

⇒ Scenario where choosing a large file system block size might be a benefit.

Benefit:- In a scenario where a system is used for storing & processing large multimedia files, such as media production company managing high resolution images & videos.

Explanation:-

Large multimedias files are typically accessed sequentially during editing, rendering or playback processes. With a large file system block size, more data can be read or written in a single I/O operation, reducing the number of disk access required to process these large files. This can lead to improved performance & efficiency when working with such data-intensive files.

Scenario where choosing a large file system block size might be a hindrance:-

In a scenario where the system is used for hosting a website with numerous small files, such as HTML, CSS & Javascript files:-

Explanation:-

Small files are common in web hosting environments & they are often accessed randomly rather than sequentially. In this scenario, a large file system block size could lead to internal fragmentation & wasted disk space. Since each small file could occupy an entire block, there would be unused space within each block, reducing the overall disk utilization efficiency.

Additionally, random access to small files may result in reading more data than necessary with a large block size.

In summary, while a large file system block size can benefit scenarios involving large sequential files like multimedia data processing, it may hinder system dealing with numerous small files & random access patterns.

Describe ACL, ACM - - -

⇒ Access Control Matrix (ACM):-

Description:-

An Access Control matrix is a formal model used to define & control access rights to objects in a system. It consists of rows representing (user, processes) & column representing objects, (files, directories). Each entry in the matrix specifies the access rights a subject has to an object.

Example:-

Consider ACM for a file system.

Or

	File A	File B	File C
User 1	R	RW	-
User 2	-	R	W
User 3	RW	-	R

In this matrix, User 1 has read access to fileA, read-write access to fileB, & no access to FileC. User 2 has read access to FileB & write access to fileC. User 3 has read-write access to fileA, read access to file C, & no access to file B.

Access Control list (ACL).

Description:- An access control list is a list of permissions attached to an object that specifies which users or system processes are granted access rights to that object. Each entry in the list corresponds to a specific user or group & defines the permissions they have on the object.

Eg:-

Using same file system.

File A :

User 1 : Read, Write

User 3 : Read, Write.

File B :

User 1 : Read, Write.

User 2 : Read

File C :

User 2 : Write

User 3 : Read.

Achieving security in files:-

Control Access:- By using mechanism like Access control lists (ACL) & ACM specified permission are assigned to users or groups, ensuring that only authorized individuals can access or modify files.

Monitor Access:- These security measures allow for tracking & monitoring of file access, enabling administrators to detect any unauthorized attempts & take appropriate action.

Prevent Unauthorized Access:-

By enforcing access controls, the system prevents unauthorized users from viewing, modifying, or deleting files. Safeguard sensitive information.

Ensure Data Integrity:- Security measures help maintain the integrity of files by controlling who can make changes, reducing the risk of unauthorized alteration.

Enhance Accountability:- By recording access activities & permissions, security mechanism promote accountability, making it easier to trace actions back to specific users or processes.

Aspect	ACM	ACL
Structure	Matrix with subjects & objects as rows/cols.	list of permission attached to each object.
Granularity	High granularity, detailed access rights	Moderate granularity, permission per user/group
Scalability	Complex & less scalable	Easier to manage & scale.
Flexibility	Detailed control over subj-obj-pairs	Simpler management of permissions on objects.
Implementation	Formal security models, ,	Gurukul OS, network security for file access

VIRUS

N

Aspects

Virus

Worm

Propagation

Requires host programs/
files to replicate.

Self-replicate without
host programs / files.

Execution

Need host program
execution to spread.

Spread independently
without host execution.

Spread speed

Slower spread rate.

Faster spread rate

Attachment

Attaches to host program
files.

Doesn't attaches - -

Impact

Can cause data corruption,
System disruption.

Can consume bandwidth,
cause system issues.