



KCS-401

Operating System



Dr. Upasana Pandey

Associate Professor

Department of Computer Sciences & Engineering

IMS Engineering College (College Code: 143)

File Concept :

- Contiguous logical address space
- Types :
 - Data
 - > numeric
 - > character
 - > binary
 - Program.

File Structure :

- None - sequence of words, bytes
- Simple record structure
 - Lines
 - Fixed Length
 - Variable length
- Complex Structures
 - Formatted document
 - Relocatable load file.
- Can simulate last two with first method by inserting appropriate control characters.
- Who decides :
 - Operating System
 - Program.

File Attributes : are the parameters used to keep track of file in OS.

- Name - Human Readable form
- Identifier - unique tag (no.) identifies file within file system.
- Type - needed for systems that support different types.
- Location - pointer to file location on device.
- Size - current file size.
- Protection - controls who can do reading, writing, executing.
- Time, date, and user identification - data for protection, security, and usage monitoring.
- Information about files are kept in the directory structure, which is maintained on the disk.
- User ID $\rightarrow$ user information.

File Operations :

- File is an abstract data types.
- Create
- Write
- Read
- Reposition within file
- Delete
- Truncate
- Open(F_i) - search the directory structure on disk for entry F_i , and move the content of entry to memory.
- Close(F_i) - move the content of entry F_i in memory to directory structure on disk.

• Open files :

- Several pieces of data are needed to manage open files : maintain Openfile table.
- File pointer : pointer to last read/write location, per process that has the file open.
- File - open count : counter of number of times a file is open - to allow removal of data from open-file table when last processes closes it.
- Disk location of the file : cache of data access information.
- Access rights : per-process access mode information.

Open File Locking :

- provided by some operating systems and file systems.
- Mediates access to a file.
- Mandatory or advisory :
 - Mandatory - access is denied depending on locks held and requested.
 - Advisory - processes can find status of locks and decide what to do.

Note : File locking used to prevent from situation like deadlock.

- Shared locks. (Read)
- Write (Exclusive lock)
for read and write both.

(10.3)

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine language program.
object	obj, o	compiled, machine language, not linked.
source code	c, cc, java, pas, asm, a	source code in various language.
batch	bat, sh	commands to the command interpreter.
text	txt, doc	textual data, documents.
word processor	wp, tex, rtf, doc	various word- processor formats.
library	lib, a, so, dll	libraries of routines for programmers.
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing.
archive	arc, zip, tar	related files grouped into one file, sometimes compressed for archiving or storage.
multimedia	mpeg, mov, rm mp3, avi	binary file containing audio or A/V information. (104)

File Access Methods :

(i) Sequential Access :

- Emulates magnetic tape operation.
 - One record is processed after other.
 - Supports following operations :
- (a) read next : Read record and move pointer to next position.
 - (b) write next : write and advance the position.
 - (c) rewind : moving back to the earlier location.
 - (d) skip n records :



(ii) Direct Access :

- It is based on disk model. It allows random access. User can jump to any record and access that record.
- Following operations are supported :

- (a) read n : reading record 'n'.
- (b) write n : writing record 'n'.
- (c) jump to record n $\Rightarrow$ jump to 10.
- (d) Query current record : used to return back to this record later.

(iii) Indexed Access : Index is created which contain a key field and pointers to the various blocks.

(10.5)

Example of Index and Relative Files

Adams	
Arthur	
Ashes	
⋮	
Smith	

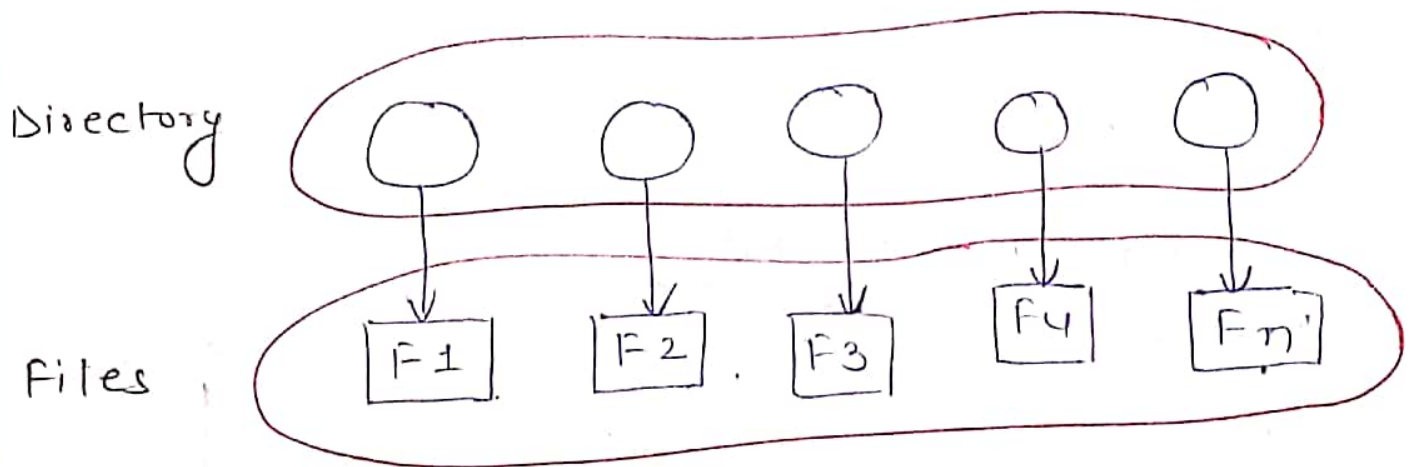
index file

Smith John	Social Security	Age

relative file

— Directory Structure :

- A collection of nodes containing information about all files.



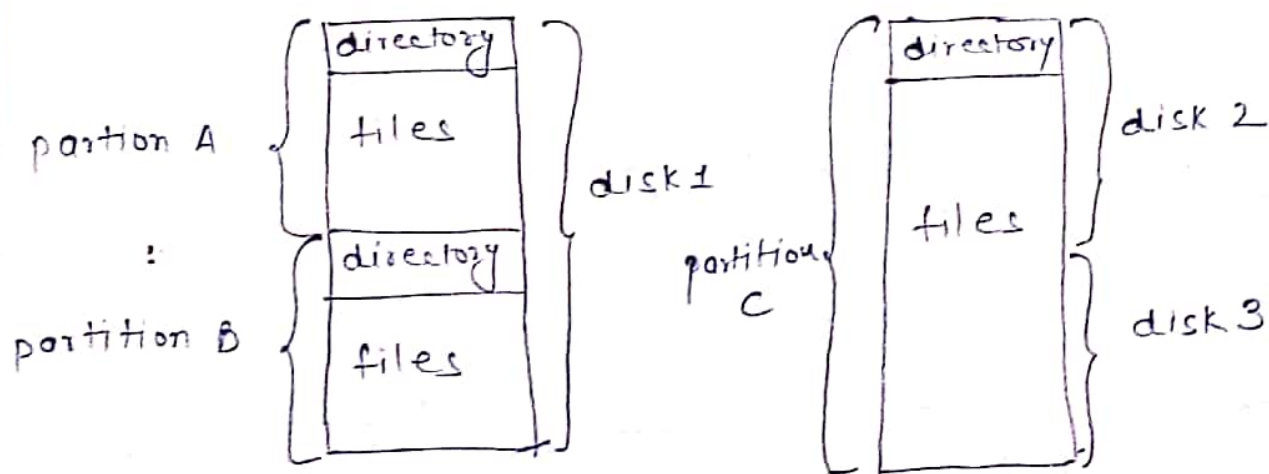
Both the directory structure and the files reside on disk.

Backups of these two structures are kept on tapes.

Disk Structure :

- Disk can be subdivided into partitions.
- Disks or partitions can be RAID protected against failure.
- Disk or partition can be used raw - without a file system, or formatted with a file system.
- Partitions are also known as minidisks, slices.
- Entity containing file system known as volume.
- Each volume containing file system also tracks that file system's info in device directory or volume table of contents.
- As well as general-purpose file systems there are many special purpose file systems, frequently all within the same operating system or computer.

A Typical File-System Organization



(10.7)

Operations Performed on Directory :

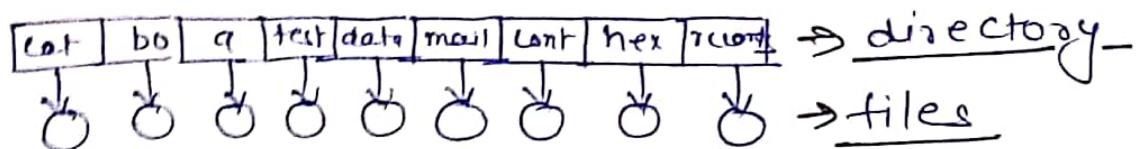
- Search for a file
- Create a file.
- Delete a file.
- List a Directory.
- Rename a file.
- Traverse the file system.

Organize the Directory (Logically) to obtain

- 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, (e.g. all Java program, all games, ...)

1. Single level Directory

- A single directory for all users.
- Each file must have unique name.



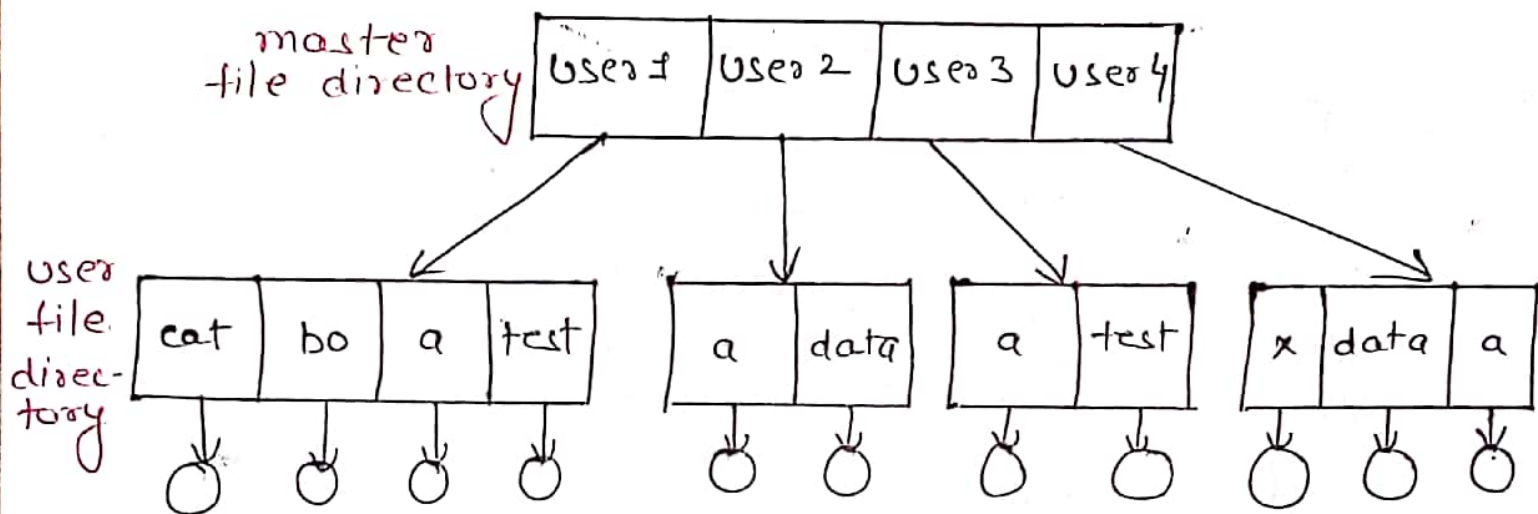
(10.8)

Naming Problem

Grouping Problem

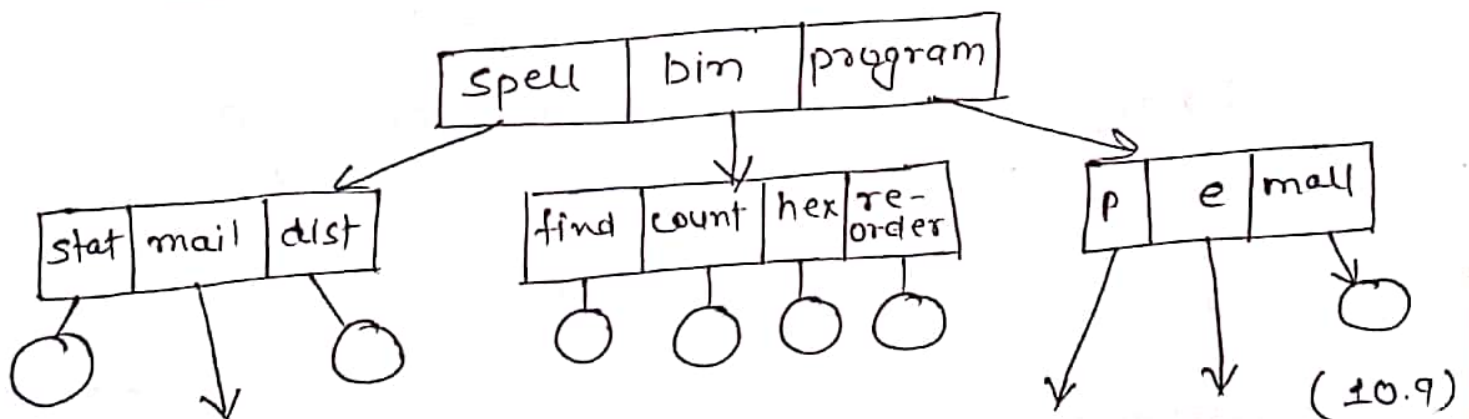
2. Two-level Directory

- Separate directory for each user



- Path name.
- Can have the same file names for different user.
- Efficient searching
- No grouping capability

3. Tree-Structured Directories



- Efficient searching
- Grouping capability
- Current directory (working directory)
 - cd/spell/mail/prog
 - type list
- Absolute or relative path name.

↓

begins at the root and follow a path down to the specified file

↘

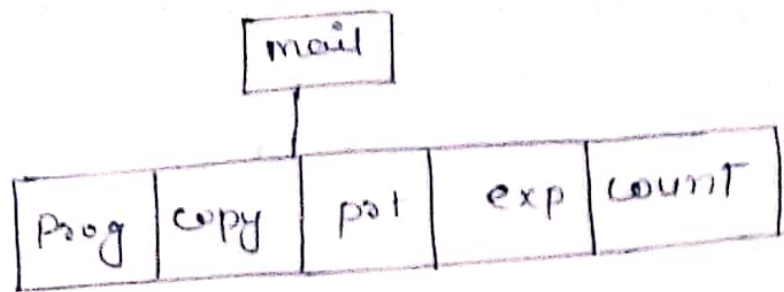
defines a path from the current directory
- Creating a new file is done in current directory
- Delete a file

`rm <file-name>`
- Creating a new subdirectory is done in current directory

`mkdir <dir-name>`

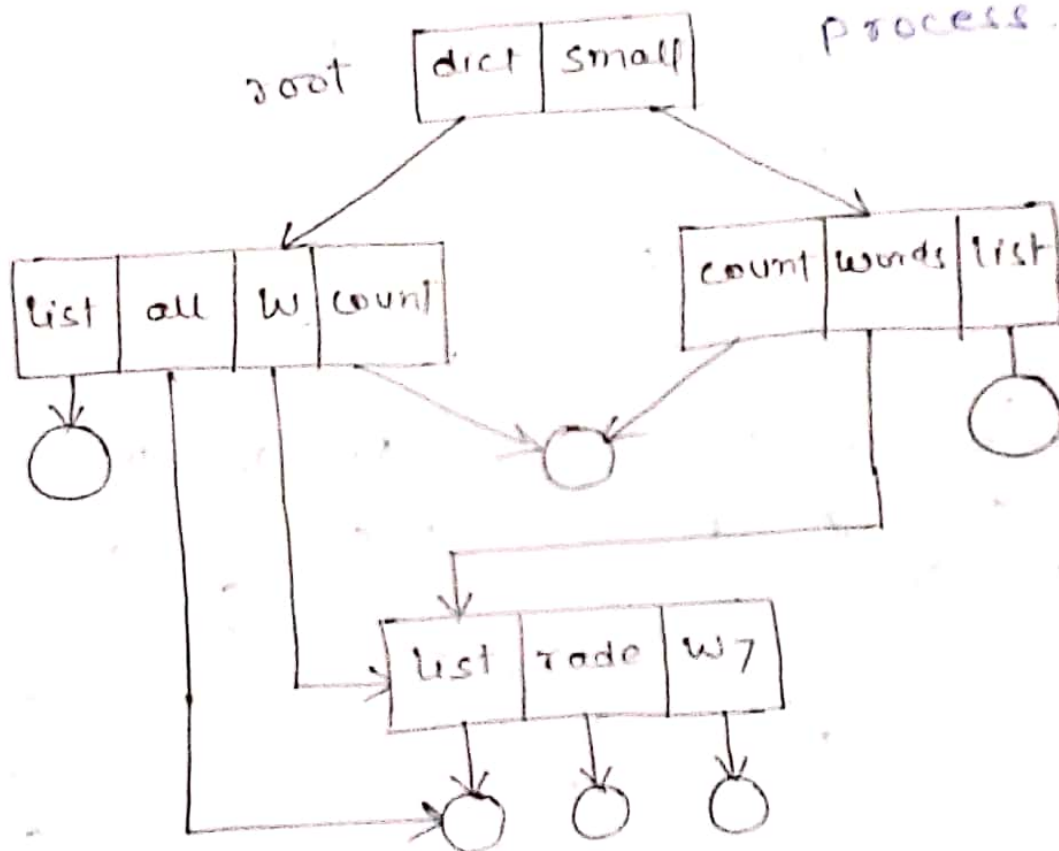
Example : if in the current directory/mail

`mkdir count`



Deleting "mail" $\Rightarrow$ deleting the entire subtree rooted by "mail".

Acyclic-Graph Directories : Files/Subdirectories shared by more than one user/process.



- Have shared subdirectories and files.
- Two different names (aliasing)
- If dict deletes list $\Rightarrow$ dangling pointers

Solutions :

- Backpointers, so we can delete all pointers variable size records a problem

(11 of 11)

File Allocation Methods :

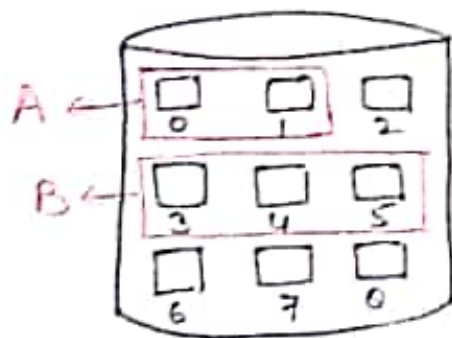
Allocate space to the files so that disk space is utilized in an efficient manner.

Factors to consider :

- (i) Processing speed $\begin{cases} \text{sequential} \\ \text{random} \end{cases}$
- (ii) Ability to use multisector and multi track transfer.
- (iii) Disk space utilization
- (iv) Main Memory requirement.

(i) Contiguous Allocation :

- Each file occupies a set of contiguous addresses on disk.



Disk

Directory.

File	start	length
A	0	2
B	3	3

- Linear ordering.
- Location of a file is defined by the disk address of the first block and its length.

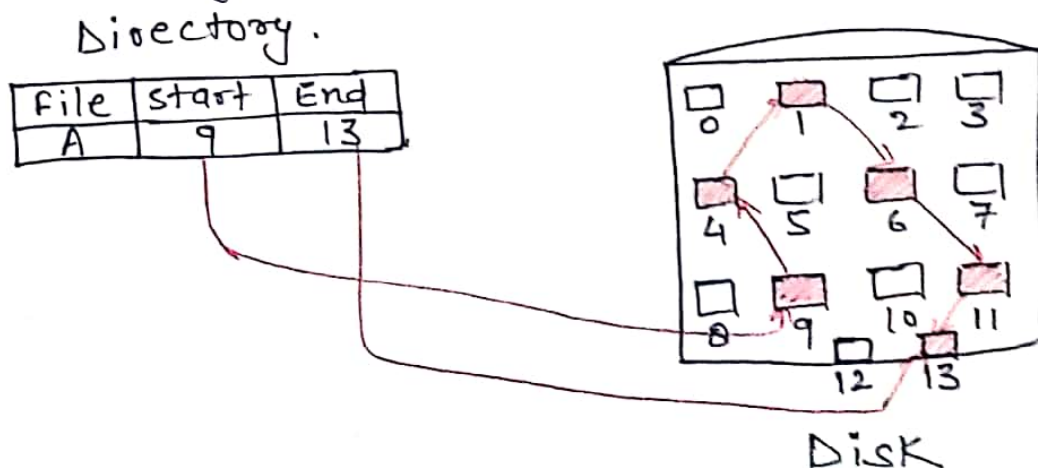
- Both sequential and direct/random access are supported.

Disadvantage :

- Finding space for new files.
- External fragmentation.
- Its hard to estimate at create time what the size of the file ultimately be. What happens when we want to extend the file - we must either terminate or try to find a bigger hole.

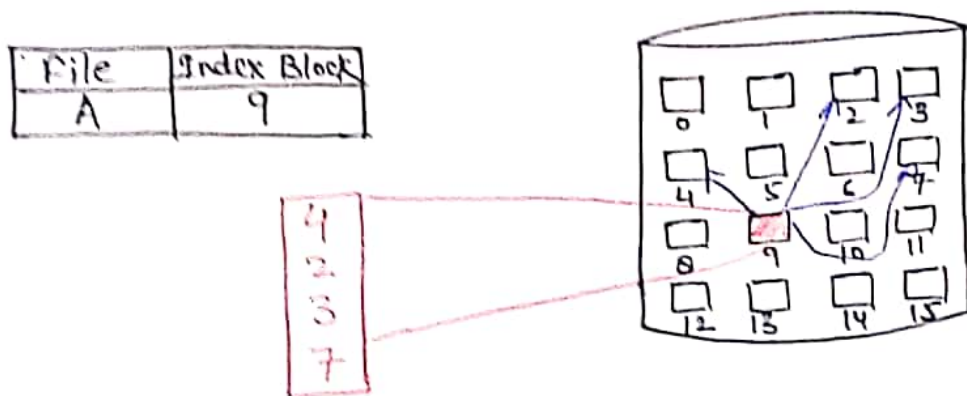
(ii) Linked Allocation :

- Solve all problems of contiguous allocation. Each file is a linked list of disk blocks.
- No external fragmentation.
- can be used only for sequential access file.



(iii) Indexed Allocation :

- Solve the problem of Linked allocation (No Random access). In this all the pointers are brought together into one location called index block. Each file has its own index block.



- Method suffers from wasted space since, for small files.
- If the index block is too small, it cannot handle the large file.

fig.