Parul® University

Operating System

Prof. Yassir Farooqui, Assistant Professor
Computer Science Engineering

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-8

I/O SYSTEMS, FILE & DISK MANAGEMENT

Topics to be Covered

- Concept of File
- Access methods
- File types
- File operation
- Directory structure
- File System structure
- Allocation methods (contiguous, linked, indexed)
- Free-space management (bit vector, linked list, grouping)
- Directory implementation (linear list, hash table)
- Efficiency and performance.

Basic File Concept

- Users data is stored in the computer Memory in terms of File.
- Permanent storage of information and data is a File.
- A file is a named collection of related information.
- File provides an easy means of information sharing.
- The operating system is not interested in the information stored in the files, operating system maps files with physical devices.
- A program which user prepare is a file. File represents program and data.
- File contains various types of data like source program, object program, text and numeric data, images, graphs, sounds, payroll records etc.

File Naming

- File store information on the disk.
- Once a file is created it is named for identification.
- File name continues to exist even after closing a file.
- Different operating systems have various notation for naming a file.
- File name contains two parts, it is separated by a period (.). For example: sorting.c
- In the file name, after period (.) is called file extension that indicates something about the file.
- The system uses the extension to indicate the type of the file and the type of operations that can be done on that file.

File types – Name, extension

File Type	Extension	Functions
Executable	exe,com,bin	Ready to run machine language program
object	Obj,o	Complied machine language
source	C,p,pass,asm,177	Source code in various language
Batch	bat,sh	Commands to command interpreter
Text	txt,doc	Textual data documents
Word	wp,nx,tef etc	Various word processor document
Archieve	arc,zip,tar	Related files group in to one,sometimes compressed

Types of file (According to internal structure of a file)

File types represents the internal structure of the file. According to the structure, file is divided into certain types

1. **Text file:** It is the sequence of characters which are organized into lines. Text file is a regular file.
2. **Source file**: It contains sequence of subroutines, procedures and functions.
3. **Executable file:** It contains a series of code sections. This file is input to the loader and loader loads the file into memory then execute.
4. **Object file:** An object file is a binary file that the compiler creates by converting the source to object code.

File types

File is divided into following types

- Ordinary file.
- Directory file.
- Device/ Special file
  - Character device file.
  - Block device file.

Ordinary file

•Ordinary file is also called regular file.
•It contains only data as a stream of characters.
•An ordinary file cannot contain another file or directory.
•It is either a text file or a binary file.

Examples of ordinary files include simple text files, application data files, files containing high level source code, executable text files and binary image file.

•**Text file:** It contains only printable characters.
•**Binary file:** It contains the printable and unprintable characters that covers the entire ASCII range.
•Regular files may be created, browsed through and modified by various means such as text editors.

Directory file

- A directory file is like a folder that contains other files,including subdirectory files.
- Directory files act as a container for the other files, of any category.
- Directory files don’t contain data,they merely contain references to the files contained within them.
- Directory is created in UNIX by the mkdir command.
- Directory is removed by using rmdir command.
- Content of directory is displayed by ls command.

Device file

- Special files are also known as device files.
- In UNIX, all physical devices are accessed via device files; they are what programs use to communicate with hardware.
- These files hold information on location, type and access mode for a specific device.
- There are two types of device files
  - **Block device file:** these files are used to access block device I/O.
  - **Character device file:** These files are associated with character or raw device access.

  Device files are created using mknod command.

File attributes

- **Name** – only information kept in human-readable form.
- **Identifier**: Every file is identified by a unique tag number within a file system.
- **Type** – needed for systems that support different types.
- **Location** – pointer to file location on device.
- **Size** – current file size.
- **Protection** – his attribute assigns and controls the access rights of reading, writing, and executing the file.
- **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.

File operations

- create
- write
- Read
- reposition within file – file seek
- delete
- open
- Close
- Truncate
- Rename
- Append

File structure

•File structure provides the mechanism for an on-line storage.

• To access the both data and program of the OS and all the users of the computer system.

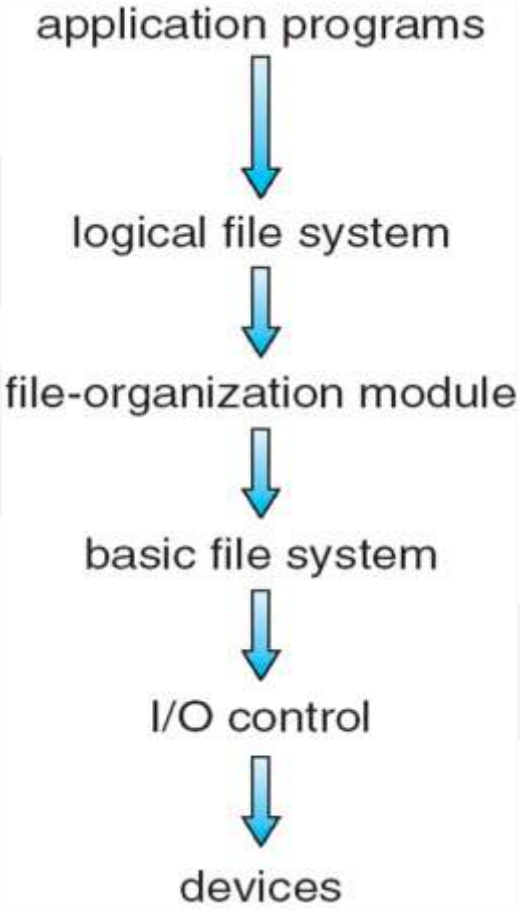
•The general file structure are as follows

- •Unstructured sequence of bytes.
- •Sequence of fixed-length records with internal structure.
- •Tree of records used with mainframe systems.

File structure

- **Unstructured sequence of bytes**
  - Most flexible.
  - Any meaning can be imposed by the user program.
  - Used by Unix and windows.
- **Sequence of fixed-length records with internal structure**
  - One entire record is read/written at a time.
  - This structure is not used currently.
- **Tree of records used with mainframe systems**
  - Records may be of varying lengths.
  - Each record has a key in fixed position in a record.
  - Records sorted by the key for easy search.

Layered File System



Parul®
University

DIGITAL LEARNING CONTENT

File access methods

- File access method defines the way process read and write files.
- Access method means accesses to files that uses a particular files.
- The file organization are implemented by an input output control system.

- Sequential access.
- Index sequential method.
- Direct/Random access.

Access Methods

Sequential Access

read next
write next
reset
no read after last write
(rewrite)

Direct Access

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

n = relative block number

Sequential access method

- Sequential access method is simple method.
- The information in a file is accessed sequentially one record after another.
- In this method, a process could read all the records in a file in order, starting at the beginning.
- A read operation reads the next portion of the file and automatically advances a file pointer.
- A write appends to the end of the file and advances to the end of the newly written material.
- Sequential access is based on a tape model of a file and works as well on sequential-access devices.
- Compiler and editor usually access files in this method.
- Transaction file is also example of sequential access method.

Disadvantages of sequential access method

- It provides poor performances.
- More efficient search technique is required.

Index sequential method

•These methods construct an index for the file.
•The index, like an index in the back of a book, contains the pointer to the various blocks.
•To find a record in the file, we first search the index and then by the help of pointer we access the file directly.
• To find an entry in the file , we first search the index and then use the pointer.
•To access the file directly and to find the desired entry of the file.

Disadvantages of Index Sequential File:

•The index file itself, may become too large to be kept in memory.
•The primary index file would contain pointers to secondary index files.

Direct access method

•It is also called random access method.
•This access allows a user to position the read/write mark before reading and writing.
• The file is viewed as a numbered sequence of block or record.
•Direct access method provides, accessing the records directly.
•Direct access method is based on the hard disk that is a direct access device.
• It allows random access of any block.
•Each record has its own address on the file by which it can be directly accessed for reading or writing.
•There is no restriction on the order of reading and writing for a direct access file.

Direct access method

•At the time of file creation, access method is defined.
• According to defined access method, file is accessed.
•Sequential access of direct file is possible but direct access of sequential file is not.
•A block number provided by the user to the OS is normally a relative block number.
•The first relative block of the file is 0 and then 1 and so on.

Disadvantages of Direct access method

•Poor utilization of input output devices.
•Consumes CPU time for record address calculation.

Directory

- Directory is a data structure that lists files and subdirectories in a collection.
- Directory does not store any user data but contains only its reference.
- File system maintain directories or folders for keeping information of files.
- User stores file in a file system using directory.
- Directory structure organizes and provides information about the files in system.
- Single dot (.) indicates a current directory and double dot (..) indicates a parent directory.
- The root is identified by the directory '/'.
- The root file system has several subdirectories.
- Window file system uses a letter followed by colon for specifying root directory.
- Directory structure contains a list of entries one for each file.

Operation on directory

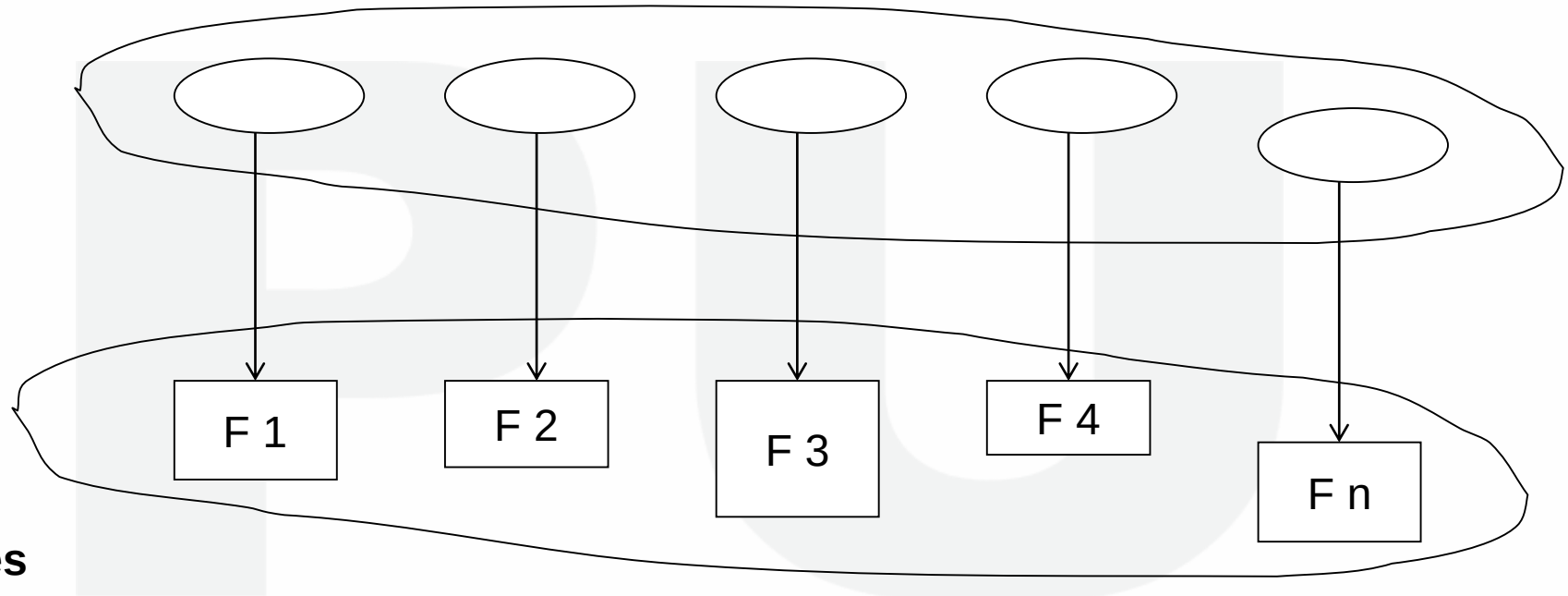
- **File searching:** directory structure is searched for particular file entry.

File uses symbolic names and similar names may indicate a relationship between files.

- **Create a file:** user create new file and added to the directory.
- **Delete a file:** user delete the file after completing its work on files.
- **Rename a file:** user change the file name if file content is change.
- **Listing of directory:** listing of file in the directory for some use.

MS-DOS and Windows uses “dir” command and Linux/Unix uses “ls” command for this purpose.

Directory Structure



Directory structure[2]

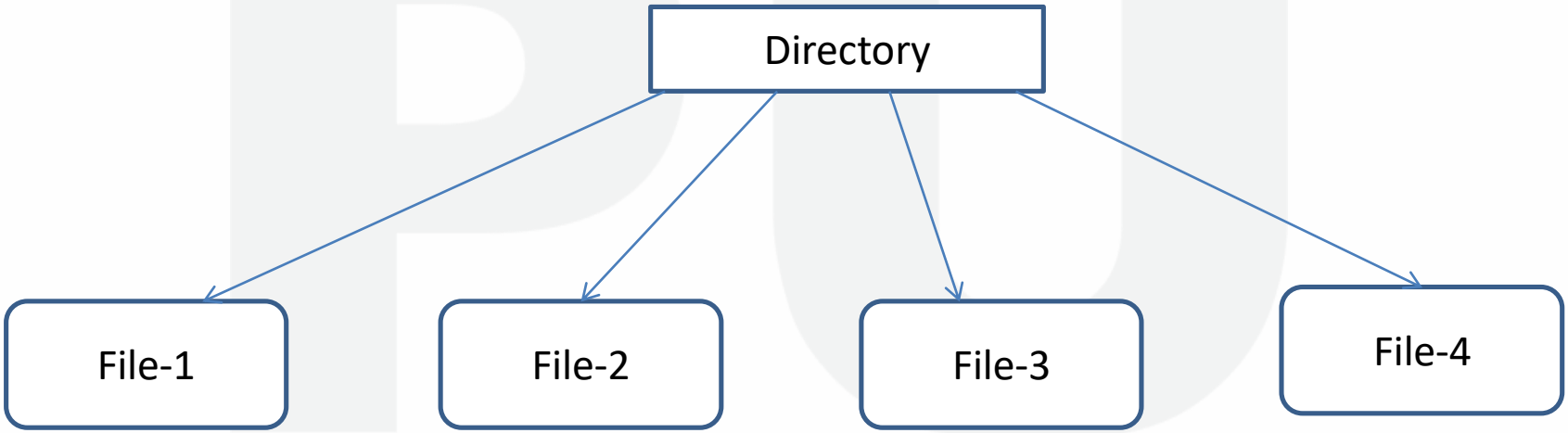
Directory structure

There are various structure of directory

1. Single level directory.
2. Two level directory.
3. Tree structured directory.
4. Acyclic graph directory.
5. General graph directory.

Single level directory

- This is simple directory structure and easy to understand.
- In single level directory all files of all users are store in one directory.

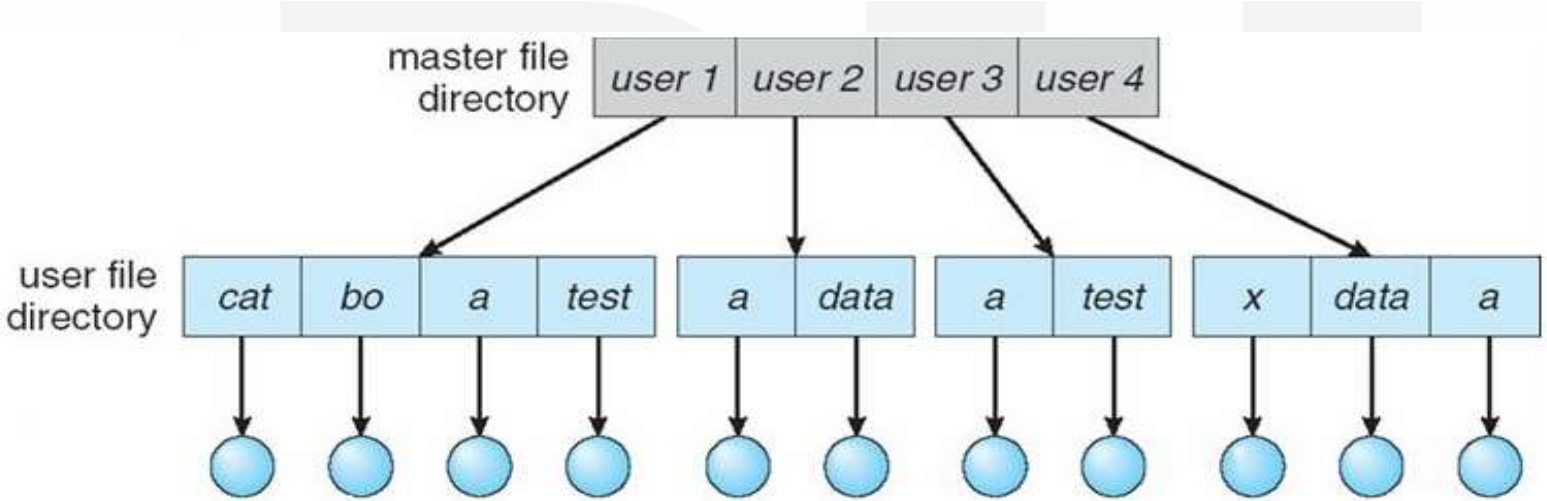


Single level directory

- Here the directory has 4 files (in fig).
- One limitation in single level directory is file naming.
- All the file name must be unique into the directory.
- If Duplicate name of the file is given by user, than unique name rule will be violated.
- MS-DOS O.S. allow 11- character and UNIX /LINUX allow 255 character for file name.

Two level directory

- The unique name problem is removing into the two level directories.
- To remove this problem each user has its own directory.



Two Level directory structure[1]

Two level directory

- In two level each user has its own directory called user file directory (UFD).
- Every UFD contain the files of only one user.
- When user will login, system will search Master file directory (MFD) by name or account number of user .
- MFD contain entry for users by name or account number and each entry point to the UFD for that user.
- Here in users UFD, filename must be unique.

Two level directory

- To create or delete the file, system searches only those users UFD.
- Here every user can work independently but can't not use or share files of whether user.
- Path into two level directories is from MFD to the file.

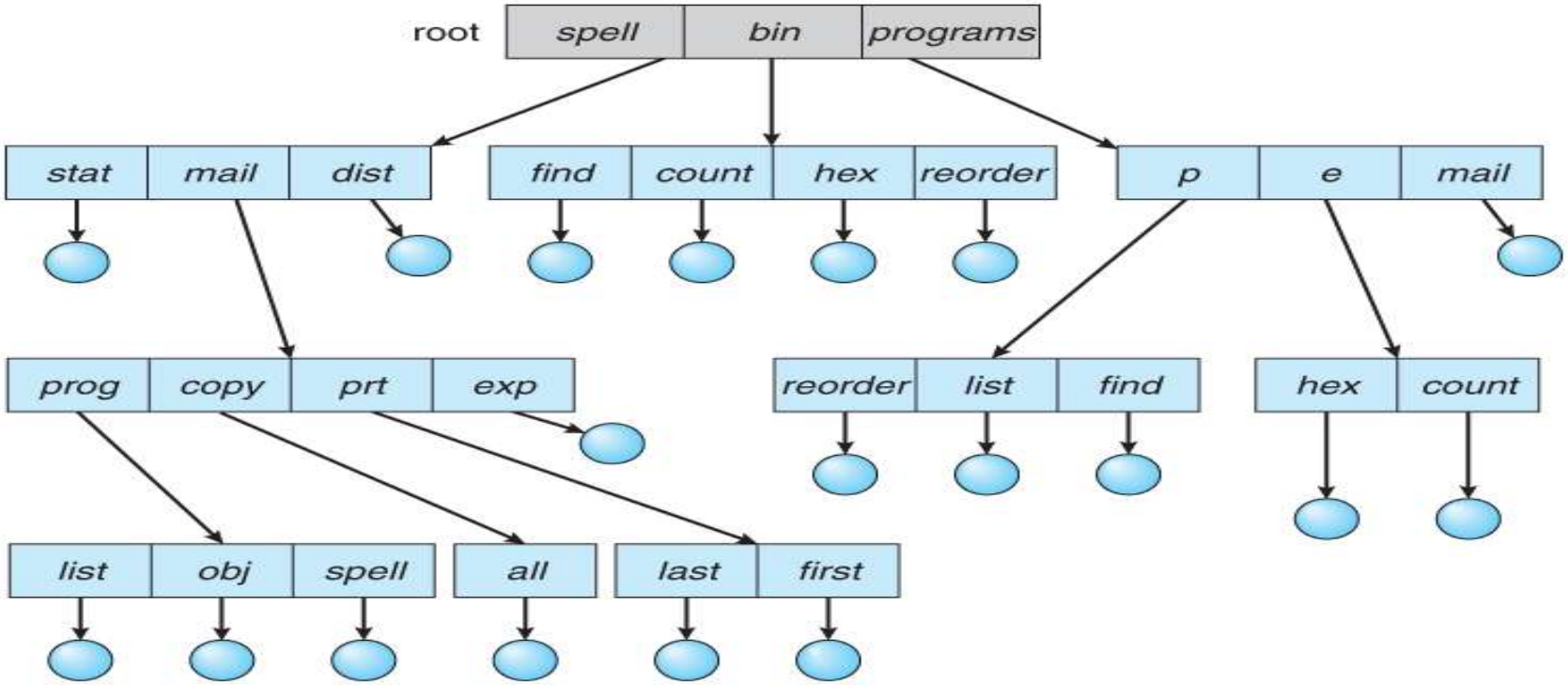
In fig. path of file-1 will be: user 1/bo/ file1.

- This structure mainly used in multi user systems.
- This structure is not suitable when user has large number of files.

Tree structure directory

- Tree structure is generalization of two level directory structure.
- In tree structure user can create is own directories, subdirectories and files.
- Tree has a root directory.
- Every file in a system has unique path from root to the specific file.
- Here directory contain sub-directories and set of files.

Tree structure directory



A Tree Structure Directory[1]

Tree structure directory

•In which directory user currently working is called current directory .

•There are two type of path :

➢**Absolute path :- path from root directory to specific file called absolute path.**

e.g. : Absolute path of file obj will be :

root / spell / mail/ proj /obj / file1

➢**Relative path :- Path from current working directory to specific file called relative path.**

e.g. : if user currently working into directory mail then relative path of file file 1 will be :

/ mail/ proj /obj / file1

Problem of deletion into tree structure

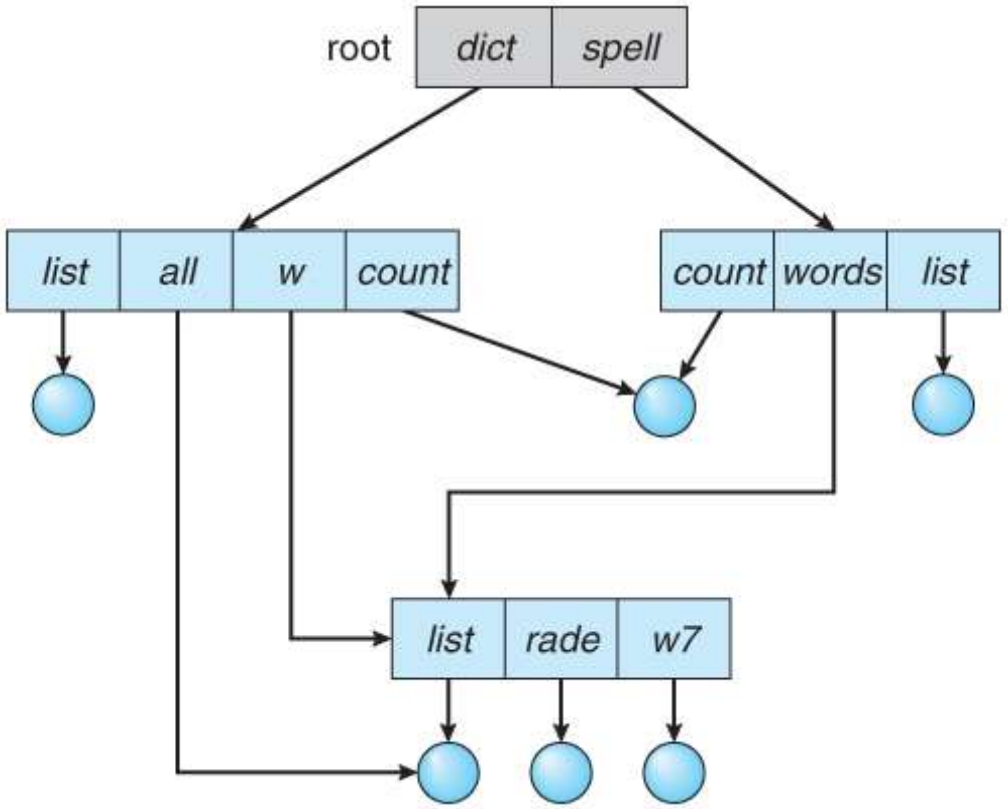
- User can not delete directly any directory or sub-directories.
- To delete directory user must has to remove all files from directory.
- If directory is not empty and contains some file or sub-directories then directory will not be removed.
- In tree structure, user can not share files or directories.

Acyclic graph structure

- Acyclic graph is slightly different from tree structure.
- Acyclic graph structure allow user to share files and sub directories.
- In acyclic graph same file or sub-directories may be link in two different directories.

Acyclic graph structure

- Acyclic graph is slightly different from tree structure.



Acyclic graph Directory[1]

Acyclic graph structure

In fig. Count directories has same file and word directories has same sub-directory list.

Disadvantages :-

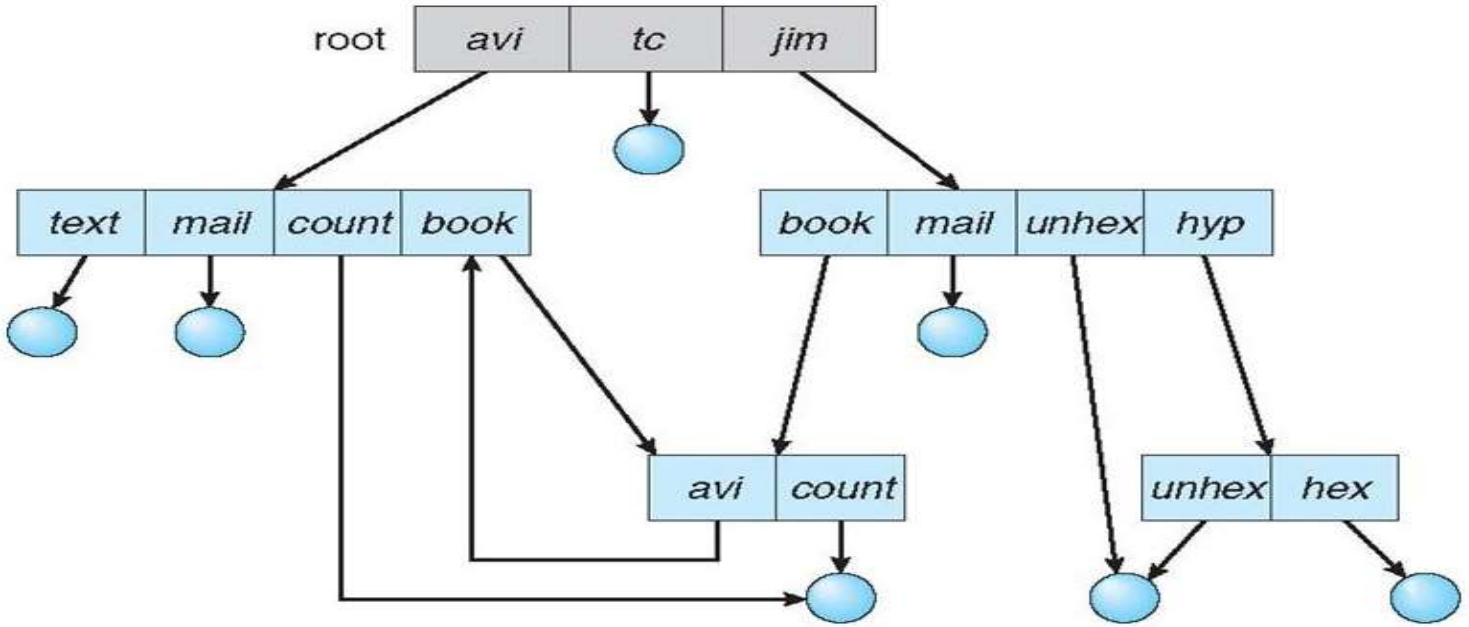
- Here several absolute path for files.
- Problem of deletion because when space allocated to shared to be de-allocated.

General graph directory

- In Acyclic graph there is no cycles.
- When link is add to the existing tree structure directory then it is called general graph directory.

General graph structure

•In Acyclic graph there is no cycles.



A Tree Structure Directory[3]

General graph directory

- Here directory is link with its subdirectory and subdirectory is link with its directory.
- In fig. avi is subdirectory of book,book is link with avi and avi is link with book.
- **HERE** ***cycle is created between book and avi.***

File sharing and protection

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

- Types of access
  - Read.
  - Write.
  - Execute.

File protection

•All data and information in computer system are stored in the form of files.
•So protection of file required from physical damage of system or improper access of file.
• Access is permitted or denied depending on several factors.

Reliability or Physical Damage

•Reliability of file provide by duplicate copies of file.
•Many system provide facility of automatic backup of files.
•File system can be damage by Hardware Problems or Power failure.

Protection of improper Access

There are various methods used for protection of file among improper access of file.

Types of Access:

- Some system permits access of file to all users, so protection is not required.
- But Some system provide free access to users, in these type of system protection is provided by **control Access.**
- In control Access method, types of file access is limited.
- Some operation on file will be controlled.
- Access Method defines which users have what privileges and permission to excel.

There are several operation may be controlled

1. Read: Read from the file.
2. Write: Write to file.
3. Execute: Execute the file.

Any other operation like Rename, Repositioning, and Truncate may be controlled.

Access list & Groups

Access List.

•**"Access list"** method provide identity dependent file access associated with each file and directory.

•Access list specify the username and type of access allowed to each user.

• when user request access to particular file, operating system check the access list associated with file.

•If user listed for requested access, than access is allowed, otherwise user can not access.

•By using the command ls –l,we can see the long listing details about the file.

Groups

Many system classify the users in 3 **group**:

1. **Owner:** The user who created the file.
2. **Group:** A Group of user who sharing the file.
3. **Universe/Others:** All other users in the system.

In this type of protection, only 3 operation are required, **Read-r ,Write-w,** And **Execute-x** .

Value assigned for these protection modes

Read(r) = 4, Write(w)= 2, Execute(x)=1

File sharing and protection

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

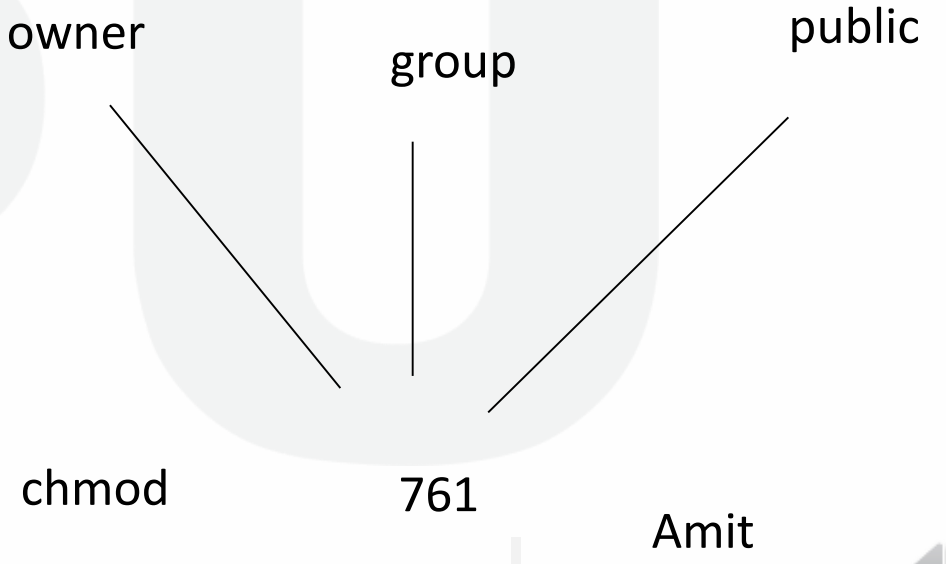
		RWX
a)Owner Access	7	1 1 1
b) Group Access	6	1 1 0
c) Others/Global	1	0 0 1

•Ask manager to create a group (unique name), say G, and add some users to the group.
•For a particular file (say *game*) or subdirectory, define an appropriate access.

File sharing and protection

- Attach a group to a file

chgrp *G* Amit



File allocation methods

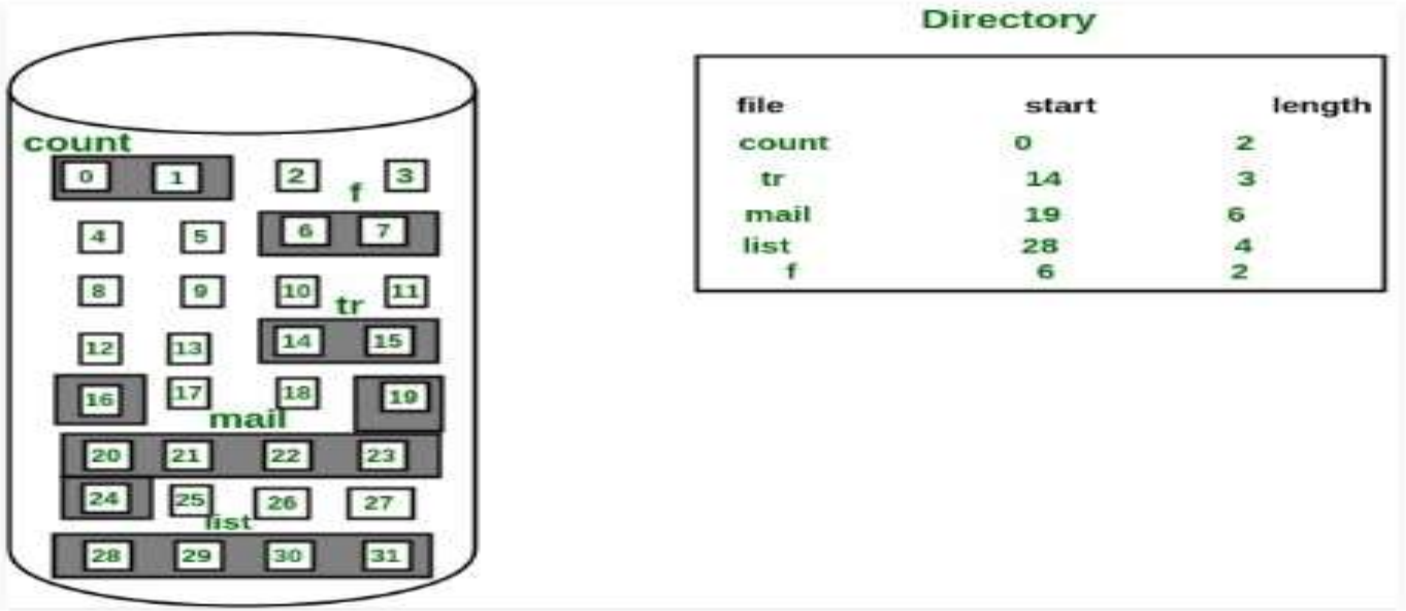
- Many files are stored on the same storage device.
- Allocation of file means allocate the space for store the file on the disk.
- Files are stored in free blocks of disk. each block size is 512 bytes.

- There are three methods are used to allocate the space for file on disk.
  - -Contiguous allocation method.
  - -Linked allocation method.
  - -Indexed allocation method.

Contiguous allocation method

- In this method, contiguously blocks are allocated to the file on disk.
- The first block is generated randomly so the block can be anywhere in memory.
- Contiguous Allocation of file define by block address and length(total number of block).
- For example, if a file requires n blocks and is given a block b as the starting location, then the blocks assigned to the file will be: *b, b+1, b+2,……b+n-1.*
- This means that given the starting block address and the length of the file (in terms of blocks required).
- we can determine the blocks occupied by the file.

Contiguous allocation method



Directory

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

Contiguous Allocation Source: Google

Contiguous allocation method

- If file size is 4 block , then continuously 4 blocks will be allocate to file on the disk.
- Each block of disk will store 512 bytes .
- The file ‘mail’ in the following figure starts from the block 19 with length = 6 blocks. Therefore, it occupies 19, 20, 21, 22, 23, 24 block.
- One disadvantage of contiguous allocation method is finding space for new files.
- In contagious Allocation directory contain **filename, start block and length**.
- The FAT table contains the name of the file,Starting location or block of file and third contains the total length of the file.
- Contagious allocation support both sequential and direct access method.

Contiguous allocation method

•As the blocks are not in continuous order it suffers from the disadvantage of Fragmentation.
•To overcome the disadvantages of Fragmentation, Compaction technique is used.
•In Compaction all the allocated blocks are grouped at the top and all the free blocks are allocated later.
•So we get continuous block of memory after Compaction.

Contiguous allocation method

Advantages

•Simple to implement
•Fast access of file

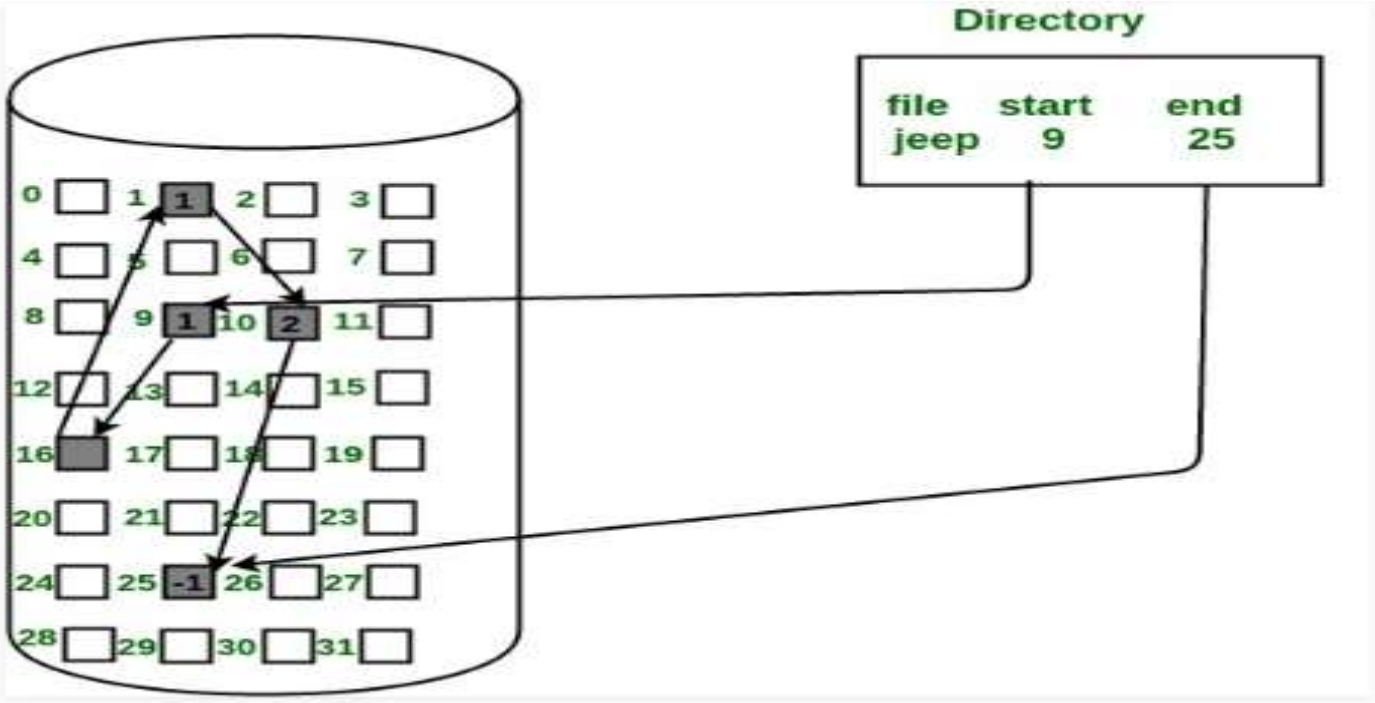
Disadvantage

•Finding the free space on disk the blocks are not in continuous order it suffers from the disadvantage of Fragmentation.
•To overcome the disadvantages of Fragmentation, Compaction technique is used.
•In Compaction all the allocated blocks are grouped at the top and all the free blocks are allocated later.
•So we get continuous block of memory after Compaction.

Linked allocation method

- Linked allocation is different from contagious allocation.
- In Linked allocation, blocks of file are linked with other blocks.
- Here blocks are allocated randomly and blocks are linked one by one.
- Each block will link with the next block of file.
- Here the directory contains **file name, start blocks number** and **ending block number.**
- In this method OS will allocate the free block randomly and link every block with next block of file.

Linked allocation method



Linked Allocation Source: Google

Linked allocation method

•The file ‘jeep’ in following image shows how the blocks are randomly distributed. The last block (25) contains -1 indicating a null pointer and does not point to any other block.

•Here all blocks are linked.

•Same procedure is for other file**.**

•Linked allocation can support only sequential access. It is not support direct access.

•Linked allocation is also called **chain allocation.**

Linked allocation method

Advantages

•Any free block can be allocated.

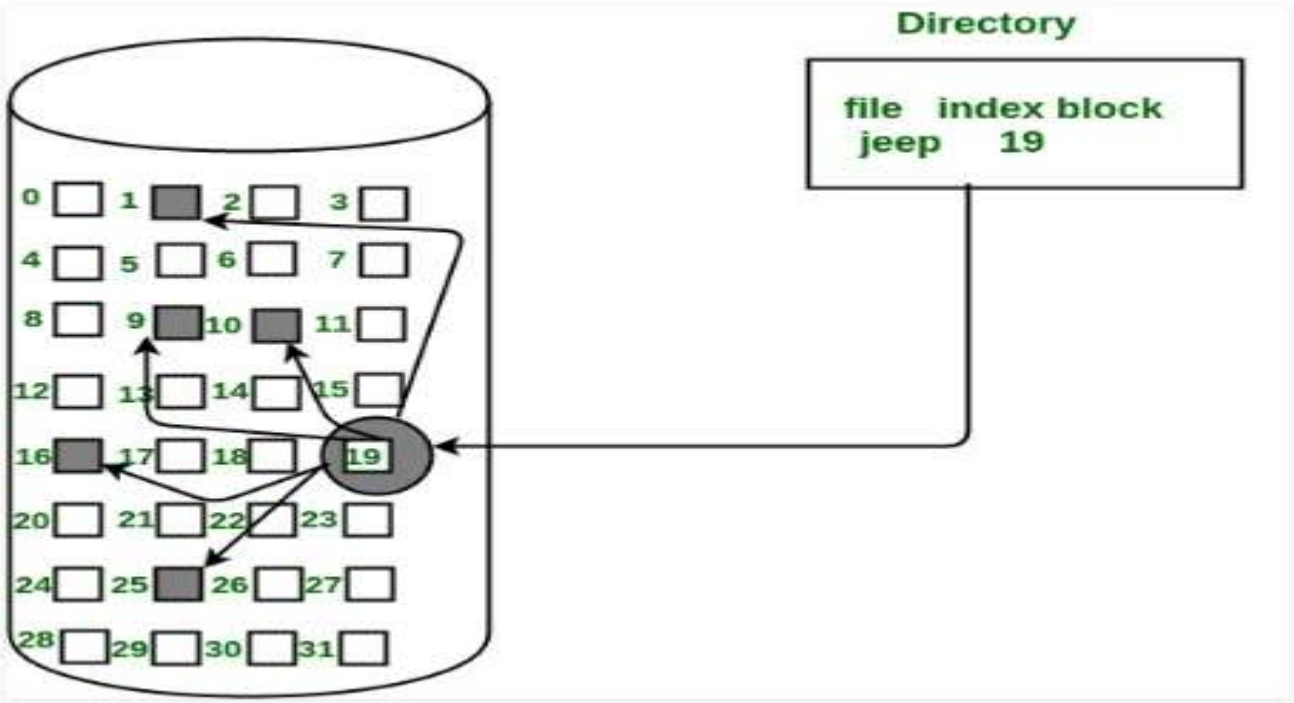
Disadvantages

•In linked allocation is each block contain the pointer to next block of file.
•4 byte is used to store the pointer of block of 512 bytes.
•It users a pointer so if a pointer field consist of error it gives error that a whole file is corrupted as maintenance of pointer is difficult.
•The major problem is that it can be used only for sequential access-files.
•So only 508 byte of data can be stored.

Indexed allocation

- In index allocation all pointer for blocks are bringing together into only one block of disk.
- In index allocation all pointers are put into one block.
- This block is called index block.
- Each file has its own index block, which contain the pointer address of all blocks of files which are allocated on disk.
- In index allocation directory entry contain the **filename** and **index block.**

Indexed allocation



Indexed Allocation Source: Google

Indexed allocation

•For example : In fig file **jeep**, has ***index block 19***, means blocks 19 will contain the pointer of all blocks of file.

•Here block 19 is containing address of block 1,9,10,16, and 25.

•Every file has only one index block. and it should be unique.

•Index allocation supports direct access.

•No more than one file has same index block.

Disadvantage

•In index allocation there are lot of waste of space into the index block.

•Because index block of 512 byte contain only address of pointer, so maximum space is waste into the index block.

Memory management / Free space management using Bitmap and Linked list allocation

- Files are created and deleted frequently by the user of Computer system.
- If the files are created and its not being used for larger time it unnecessary uses an memory.
- Since disk space is to be allocated to files.
- The free space available must be managed so as to allocate space to the new files being created.
- To retain the information about the free disk space, the system maintains a free-space list.
- The free-space list keeps information about the free disk blocks.
- Two methods are used for free memory management
    1) Bit Map
    2) Linked List

Memory management by Bitmap

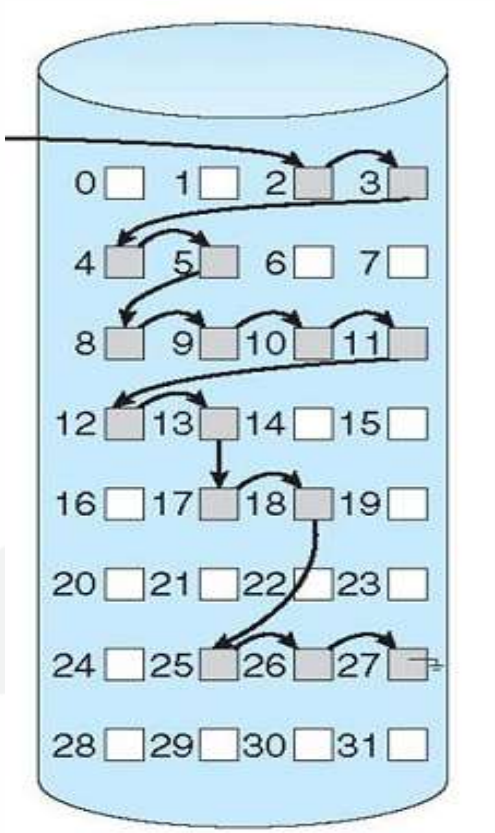
- This method represents the allocation status of disk block as a bit map or bit vector.
- Each block is represented by 1 bit. If the block is free, the bit is 0; if the block is allocated, the bit is 1.
- For example, consider a disk with 30 blocks.
- The bit map can be: 000001111001111110001100000011
- If the blocks 2,3,4,5,8,9,10,11,12,13,17,18,25 are free then
  001111001111110001100000010000…….
- The main advantages of this method is its relatively simple and n contiguous free blocks that can be found together.
- Each time a new block is to be allocated to a file, the bit vector will be updated respectively.

Memory management by Linked list

•In this method of free space management, all the free blocks are linked with each other.
•The pointer referencing to the first free block is placed in a special location on the disk and cached in memory.
•This first block holds an address pointing to the next free disk block and so on.
•However, this method is not efficient because traversing of the list requires the reading of each block, which consumes more I/O time.

Memory management by Linked list

•The free space list given in figure is as follows; a pointer is stored for block 2, as the first free block.
•Block 2 would contain a pointer to block 3, which would point to block 4 and so on.



Memory Management by linked list[3]

File System implementation

- There are 2 methods to implement file system depend on the types of the operating system
  - On-disk structure.
  - In-memory structure.

On-disk structure for file implementation

•On disk, file system may contain information about how to boot an operating system stored there.
•The total number of blocks, the number and location of free blocks, the directory structure and individual files.
•Some of these structures are described as follows:
Boot control block(per volume): It can contain information needed by the system to boot an operating system from that volume.
• If disk does not contain OS, this block can be empty.
•This is typically the first block of a volume.
•In UFS, it is called **boot block**; in NTFS it is called **partition boot sector.**

On-disk structure for file implementation

•**Volume control block(per volume)**

• It contains volume (or partition) details, such as number of blocks in a partition.
•The size of block, a free block count, a free block pointers and free FCB count and free FCB pointers.
•In UFS, it is called **superblock**; in NTFS, it is stored in **master file table.**
•**A directory structure (per file system)** is used to organize the files.
• In UFS, this include file names and associated inode number.
• In NTFS, it is stored in master file table.
•**A per-file FCB(File control block)** contains many details about the file.
•It has unique identifier number to allow association with a directory entry.

Typical FCB is shown in figure

•In NTFS, this information is actually stored within master file table which uses relational database structure, with a row per file.

file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks or pointers to file data blocks

Contiguous Allocation Source: Google

In-Memory structure of File implementation

•The in-memory information is used for both file system management and enhancing the performance through caching.
•This structure includes the following:
•**In-memory partition table:** it contains the information about each mounted partition.
•**An in-memory directory structure:** This directory structure cache holds the directory information of recently accessed directories.
•**System-wide open-file table:** It is used to maintain a copy of FCB of opened file along with its other information.
•**Per-process open-file table:** it has a pointer to the suitable entry in the system-wide open-file table along with other information.

Directory Implementation

•The selection of directory-allocation and directory management algorithm significantly affects the efficiency, performance and reliability of the file system.

•The different organizations are:

Liners list.

Hash table.

Directory implementation using linear list

•The simplest method of implementing a directory is to use a linear list of file names with pointers to the data blocks.
•This method is simple to program but time consuming to execute.
•To create a new file, we must first search the directory to be sure that no existing file has the same name.
• Then ,we add the new entry at the end of directory.
•To delete a file, we search the directory for the named file and then release the space allocated to it.

Directory implementation using linear list

- The real disadvantage of the linear list of directory entries is that finding a file requires a linear search.
- Directory information is accessed frequently, thus in many Operating System.
- A software cache is used to store the most recently used directory information.
- If a sorted list of file names is maintained, then a binary search can be used to decrease the average search time.
- Other data structure like B-Tree might be of an advantage in this type of a case.

Directory implementation using hash table

- Hash table is the another method for implementing directory structure.
- With this method, a linear list stores the directory entries, but hash data structure is also used.
- The hash table takes a value computed from a file name and return a pointer to the file name in the linear list.
- It can greatly decrease the directory search time.
- Insertion and deletion are also fairly straight forward although some provision must be made for collision.
- Situation in which two file names hash to same location.

Directory implementation using hash table

- To resolve this problem, chained-overflow hash table is used.
- Each hash entry can be linked list instead of an individual value.
- It means if the hash values point to the same location, a chain of linked list is formed.
- So, in this way we can remove the collisions

File system recovery

- There are many causes that are responsible for computer crashes or physical damage.
- In this situation, file systems are expected to behave sensibly.
- These problem may result in file system consistencies.
- In order to check consistency, we need to observe the following:
  - All data and control structure, such as descriptors and bitmaps should be in an appropriate arrangements.
  - When a system reboots, it should return to a valid state.
  - It at the time of crash, a write operation is in progress.
  - Then the operation is either finished or rollback completely. This is called atomicity.

File system recovery

•**Consistency checker:** At the time of reboot, a program called consistency checker, is executed in order to check and correct disk inconsistencies.

•**The consistency checker** matches the data available in the directory structure with the data stored in blocks of disk.

• If it finds any inconsistencies, then it tries to fix it.

Methods of file system recovery

- **Backup and restore.**
- **Log-structured file system.**

Backup and recovery

•When the magnetic disks fails, a backup of disk may be taken on another storage device in order to restore the system.
•For minimizing the copy, the time when the last file was modified and the backup taken can be considered.
•The file's last write date in the directory structure specifies that the file has not changed since that date.
•If Not then it is not required to copy that file again.

Log structured file system

•The main purpose of using the log structured file system is to enhance the write performance.
•It buffers a series of file system changes in the file cache and them write all such changes to the disk sequentially in a single write operation.
•Once the changes are written to the log, user process are allowed to continue execution.
•It transaction is completed, it removes the log files.
•If the system crashes, there will be zero or more transactions in a log file.
•Therefore they must be completed, and the file system structure remain consistent.

Virtual file system

•A virtual file system (VFS) is programming that forms an **interface between an operating system's kernel and a more concrete file system.**

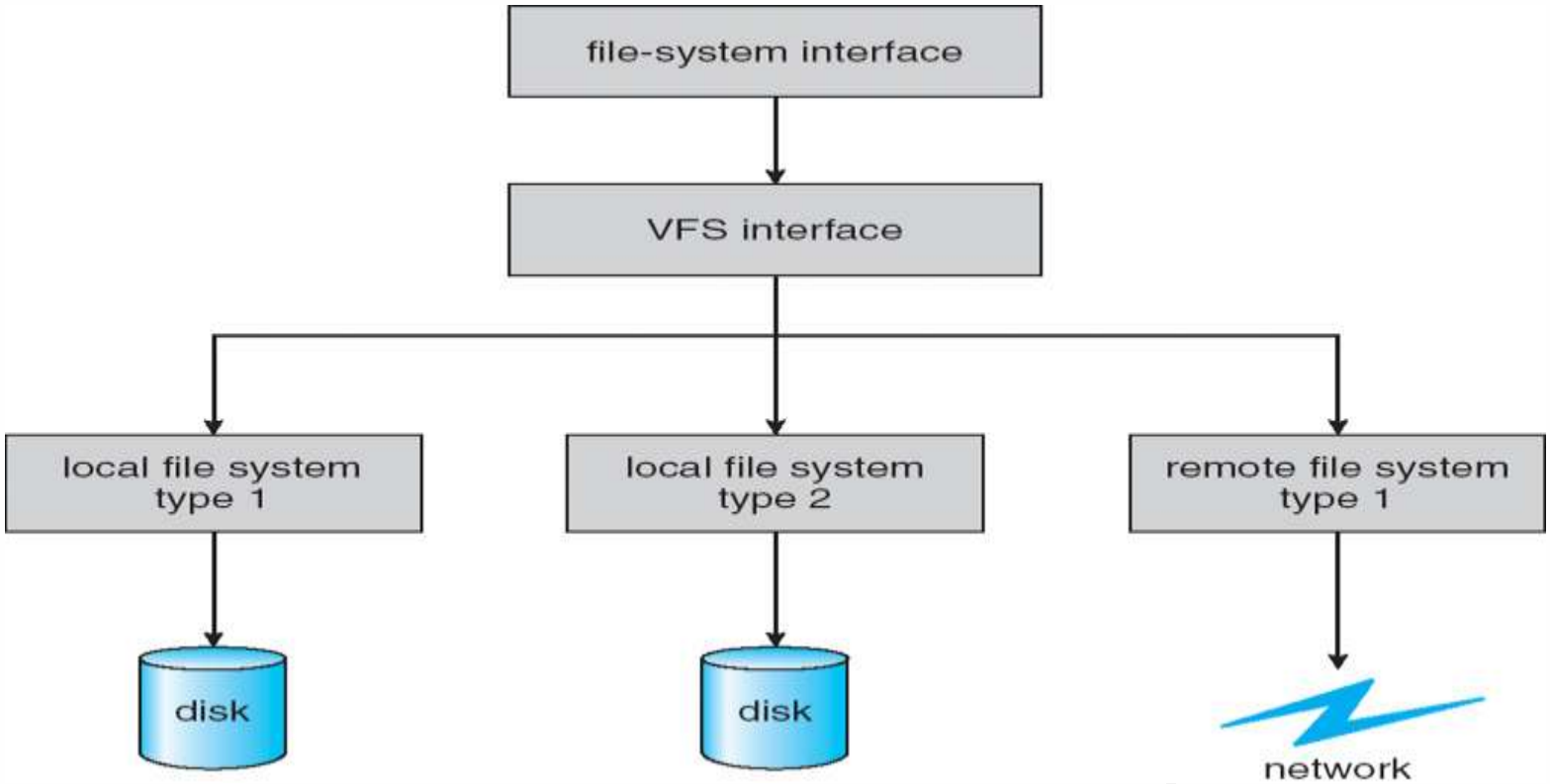
•The purpose of a VFS is to allow client applications to access different types of concrete file systems in a uniform way

•A VFS can, for example, be used to access local and network storage devices transparently without the client application noticing the difference.

•It can be **used to bridge the differences in Windows, Mac OS and Unix filesystems.**

•So that applications can access files on local file systems of those types without having to know what type of file system they are accessing.

Virtual file system



Virtual File System[3]

File system performance

•File System performance is often a major component of overall system performance.
•It is heavily dependent on the nature of the application generating the load.
•To achieve optimal performance, the underlying file system configuration must be balanced to match the application characteristics.
•Once we have a good understanding of the application, we can try and optimize the file system configuration.
•To make the most efficient use of the underlying storage devices.

Improve file system performance

- Reduce the number of I/O's to the underlying device(s) where possible.
- Group smaller I/O's together into larger I/O's where possible.
- Optimize the seek pattern to reduce the amount of time spent waiting for disk seeks.
- Cache as much as data as realistic to reduce physical I/O's.

Refrences

[1]. Operating System Concepts - 9th Edition by Abraham Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
[2]. Operating Systems: Internals and Design Principles - 5th Edition, William Stallings, Prentice Hall of India
[3]. Modern Operating Systems- 4th Edition Andrew S. Tanenbaum, Pearson Education India
[4]. Operating Systems: A Modern Perspective - 2nd Edition by Gary J. Nutt, Addison-Wesley
[5] Design of the Unix Operating Systems -8th Edition by Maurice Bach, Prentice-Hall of India.
[6] File System Concepts.
[7] Operating System File systems. Tutorialspoint. https://www.tutorialspoint.com/operating-system/file_system

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in