File Management

File:
- A file is a named collection of related information that is recorded on secondary storage such as magnetic disk, magnetic tapes
- File is a sequence of bits, bytes, lines or records whose meaning is defined by File Creator & User

Extensions → Types
- GIF → Image
- HTML → Web page
- OBJ, O → Object
- EXE, BIN → Executable
- TXT, DOC → Text
- LIB → Library
- ZIP → Archive
- PS → Post Script
- MP3, MPG → Multimedia
- PDF → Portable Document Format

mega bytes, mega bytes, mega bytes

File Structure

Unstructured Sequence of Bytes
- OS treats file as sequence of bytes. Any meaning must be imposed by user level prgm
- Both UNIX & WINDOWS use this approach
- It provides more flexibility
- Any type of content can be kept in file as per convenience

Sequence of fixed length record
- File is treated as sequence of fixed length record
- Each record has its internal structure. Any read op^n on file returns one record and and write op^n will append or overwrite one record

Tree Structure
- Files are organised in tree structure
- All records not necessarily have same length
- Each record has a key field in fixed position
- Records in tree are arranged in sorted order to key field, to permit quick searching for a particular key
- Main aim is to search record on particular key

File types

Regular Files
- It contains user info
- The byte sequence, record sequence & tree structure are examples of regular files

Directories
These are system files for maintaining the structure of file system

Character Special files
- related to I/P o/P and used to model serial I/O device such as terminals, printers...

Block Special Files
- These are used to model disks

- All regular files are normally either ASCII files or binary files

ASCII file contains lines of text, each line is terminated by either carriage return character or line feed character

File attribute

Attribute
- Protection
- Size
- Type
- Creator
- Owned
- Password
- Identifier
- Hidden Flag
- Read only Flag
- ASCII / Binary Flag
- Lock Flag
- Key length
- Time of last access
- Time of last change
- Maximum Size
- System Flags

File operations
- Create
- Delete
- Open
- Close
- Read
- Write
- Append
- Seek
- Get attribute
- Set attribute
- Rename

File Access
- Sequential access
- records are accessed in some sequence
- info in file is processed in order, one record after the other
- Such method is the most primitive one

eg: compiler usually access files in fashion
- direct or random access
- It provides accessing the records directly
- each record has its own address on the file with by help of which, it can be directly accessed for reading or writing
- Records need not be in any sequence within the file & need not be in adjacent location on storage medium

Indexed Sequential Access
- This mechanism is built up on base of sequential access
- An index is created for each file which contains pointers to various block
- Index is searched sequentially & its pointer is used to access the file directly

Directory Operations
- Create
- Delete
- Opendir
- Closedir
- Readdir
- Rename
- Link
- Unlink

File Organization

Pile
- simplest form of file organization
- File data is stored in order of arrival
- each record in the file comprises one burst of data
- idea behind pile is simply to collect mass of data & save it

Sequential File
- most commonly used structure in records have fixed formats
- Each and every record in file has same length with equal no of fixed length field in a particular order
- as length & location of each field are known simply the values of fields are required to be stored

Indexed Sequential file
- It eliminates drawback of sequential file
- it contains index to the file to support random access & an overflow file
- offers searching ability to search speedily the required record

Indexed File
- The sequential File & Indexed Sequential file can searched the record based on single field of file
- Apart from using key field, they cannot search record using other attribute of record
- To overcome this difficulty and accomplish the flexibility, a structure containing multiple indexes, one for each type of field is required
- Records are accessed only through indexes

The direct or Hashed file
- makes use of ability found on disks to access straight any block of a known address
- hashing op^n on key value using hash function
- direct files are frequently used where very fast access is necessary, where there use of fixed length records and where records are always accessed one by one

File Sharing
- Necessary to share a file among multiple users
- Needs to deal with 2 issues

① Access Rights
- None
- Knowledge
- Execution
- Reading
- Appending
- Updating
- Changing Protection
- Deletion

② Simultaneous Access
- When more than 1 user granted access to append or update a file, the OS or file management system must implement some way to restrict it

File Allocation
On secondary storage, a file consists of collection of blocks
- OS or file management system is responsible for allocating blocks of files
- Approach taken for file allocation may influence the approach taken for free space management
- Space is allocated to a file as one or more portions

File Directories

Single level directory system
- The simplest method is to have one big list of all the files on the disk
- The entire system will contain only one directory which is supposed to mention all the files present in file system.
- The directory contains one entry per each file present on file system
- Adv → Simple implementation
- Adv → We cannot have 2 files with same name
- Disadv → Hectic Searching

Two level directory system
- Separate directory for each user.
- 1 master directory which contains separate directories dedicated to each user.
- System doesn't let a user to enter in the other users directory without permission

Characteristics
- Each file has a pathname as → /user name / directory name/
- Different user can have same file name
- Efficient searching

Adv → Solve name collision pb
→ Independent user get isolated from each other

Hierarchical directory system
It is quite general and advantageous for users to group their files together in logical ways
- Adv → With this system, in addⁿ to their files, Users can access the files of other user by pathname

Path names → Absolute Path name
→ Relative Path name

Free Space Management
- A file system is responsible to allocate the free blocks to the file therefore it has to keep track of all the free blocks present in the disk

Methods of free Spaces

→ Bit Vector
- In this approach, the free space is implemented as a bit map vector. It contains the no. of bits where each bit represents each block
- If block is empty → Bit is 1 otherwise zero
- Initially all blocks are empty

→ Linked List
- This approach suggests linking together all the free blocks and keeping a pointer in the cache which points to the first free block.
- All free blocks of HD → linked together → pointer

Preallocation and dynamic allocation
- Preallocation policy requires that the max size of file declared at a time of file creation request
- For many appⁿ it is difficult to estimate reliably the max potential size of file
- Dynamic allocate space to the file portions as needed

Portion Size
- Choosing a portion size, there is a trade off bet^n efficiency from the point of view of a single file v/s Overall system efficiency

File allocation methods

Contiguous Allocation
- Each file occupies a contiguous address space on disk
- Assigned disk address is in linear order
- Easy to implement
- External fragmentation is a major issue.

Linked list allocation
- Each file carries list of links to disk blocks
- Directory contain link / pointer to first block of file
- No external fragmentation
- Effectively used in sequential address
- Ineffecient in use of direct access

Indexed Allocation
- Provides sol^n to problem of contiguous & linked allocation
- Index block is created having all pointers of file
- Each file has its own index block which stores the address of disk space occupied by file

I - Node (Index - Node)
- All types of UNIX files are administered by OS by means of index nodes
- An inode is a control structure that contains key info needed by OS for particular file

Linked list allocation using a table in memory

Linux Virtual File System
- The linux virtual file system or VFS generally is a layer that sits on the top of your actual file system which allows the user to access diff types of file system
- Linux of VFS is an interface bet^n kernel and actual file system
- Object oriented principles are used to design VFS
- 4 major object types by VFS
  - → Superblock
  - → dentries
  - → inodes
  - → files
- A set of operations are defined for each of the type of objects.
- Each object of one of these types points to f^n table.
- The record of addresses of the actual f^n are kept in f^n table