

Files:

- Data collections created by users.
- The file system is one of the most important parts of the OS to a user.

Properties of Files:① Long term existence:

- Files are stored on disk or other secondary storage or do not disappear when a user logs off.

② Sharable between processes:

- Files have names and can have associated access permissions that permit controlled sharing.

③ Structure:

- Files can be organized into hierarchical or more complex structure to reflect the relationships among files.

File structure:① Unstructured sequence of bytes:

- OS treats content of the file as sequence of bytes. Any meaning must be imposed by user level program.
- Both UNIX and Windows use this approach.
- It provides more flexibility.
- Any type of content can be kept in the file as per user's convenience.

② Sequence of fixed length record:

- File is treated as sequence of fixed length record.
- Each record has an internal structure. Any read operation on file returns one record and write operation will append or overwrite one record.

③ Tree structure:

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

File Types:① Regular Files

- It contains user information.
- The byte sequence, record sequence and tree structured are the examples of regular files.

② Directories

- These are system files for maintaining the structure of file system.

③ Character special file

- These files are related to input/output and used to model serial I/O devices such as terminals, printers and networks.

④ Block special files

- These are used to model disks.
- All regular files are either ASCII files or binary files.
- ASCII file contain lines of text. Each line is terminated by either carriage return character or line feed character.

File Attributes

- File is identified by its name which is string of the character.
- Every file has a name and its data.
- All OS associate other information with each file.

Example - The date and time of file creation and the size of the file.

- Such extra information associated by OS is called as attributes.
- Attributes list varies from one OS to another.

① Protection

- Access-control information, determines who can do reading, writing, executing and so on.

② Size

- The current size of the file.

③ Type

- Information is needed for systems that support different types of file.

④ Owner

- Current owner of the file.

⑤ Password

- Password needed to access the file

⑥ Identifier

- The unique no. which identifies the file within the file system

⑦ Hidden Flag

- 0 for normal
- 1 for do not display in listings

⑧ Read only flag

- 0 for read/write
- 1 for read only

⑨ ASCII/Binary Flag

- 0 for ASCII file
- 1 for Binary file

⑩ Lock Flags

- 0 for unlocked
- Non zero for locked

⑪ Key length

- No. of bytes in the key field

⑫ Time of last access

- Date and time the file was last accessed

⑬ Time of last change

- Date and time the file has last changed

⑭ Maximum size

- No. of bytes the file may grow to

⑮ System Flag

- 0 for normal file
- 1 for system file

File Operations:

① Create

② Delete

③ Open

④ Close

⑤ Read

⑥ Write

⑦ Append

⑧ seek

⑨ Get Attributes

⑩ Set Attributes

⑪ Rename

These are common system calls relating to files.

File Access

① Sequential Access

- Most of the OS access the file sequentially.

In sequential access, the OS read the file word by word. A pointer is maintained which initially points to the base address of the file. If the user wants to read first word of the file then the pointer provides that word to the user and increases its value by 1 word. This process continues till the end of the file.

② Random or Direct Access

- Mostly required in the case of database systems. In most of the cases, we need filtered information from the database..

Suppose every block of the storage stores 4 records and we know that the record we needed is stored in 10th block, the sequential access will not be implemented because it will traverse all the blocks in order to access the needed record.

Direct access will give the required result despite of the fact that the OS has to perform some complex tasks such as determining the desired block number.

③ Indexed Access

- If a file can be sorted on any of the fields then an index can be assigned to a group of certain records

However, a particular record can be accessed by its index.

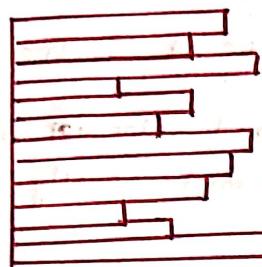
The index is nothing but the address of a record in the file

In index accessing, searching in a large database became very quick and easy but we need to have some extra space in the memory to store the index value.

File Organizations

① The File

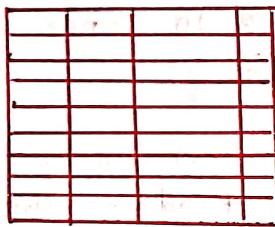
- Simplest form of file organization
- Data are collected in the order they arrive
- Each record consists of one burst of data.
- Purpose is simply to accumulate the mass of data and save it.
- Record access is by exhaustive search



- Variable length records
- Variable set of fields
- (Chronological) order

② The Sequential File

- Most common form of file structure
- A fixed format is used for records
- Key field uniquely identifies the record
- Typically used in batch application.
- Only organization that is easily stored on tape as well as disk



- Fixed-length records
- Fixed set of fields in fixed order
- Sequential order based on key field

③ Indexed - Sequential File

- Adds an index to a file to support random access
- Adds an overflow file
- Greatly reduces the time required to access a single record.
- Multiple levels of indexing can be used to provide greater efficiency in access

④ Indexed File.

- Records are accessed only through their indexes
- Variable length records can be employed
- Exhaustive index contains 1 entry for each record in the main file
- Partial index containing entries to records where the field of interest exist

- Used mostly in applications where timeliness of information is critical.
- Examples would be airline reservation systems and inventory control system

⑤ The direct or hashed file

- Address directly any block of a known address
- Makes use of hashing on the key value
- Often used where
 - Very rapid access is required
 - Fixed length records are used
 - Records are always accessed one at a time
- Examples
 - Directories
 - Pricing Table
 - Schedules
 - Name lists

Directory Structure

- The listing of the related files on the disk. The directory may store some or the entire file attributes.

- Common methods for defining the logical structure of a directory

- ① Single level directory system
- ② Two level directory system
- ③ Hierarchical directory system

① Single level directory

- 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 the file system. The directory contains one entry per each file present on the file system.

Advantages

- ① Implementation is very simple
- ② If the sizes of the files are very small then the searching becomes faster.
- ③ File creation, searching, deletion is very simple since we have only one directory

Disadvantages:

- ① We cannot have 2 files with the same name.
- ② The directory may be very big therefore searching for a file may take so much time.
- ③ Protection cannot be implemented for multiple users.
- ④ There are no ways to group same kind of files.

② Two level Directory

- We can create separate directory for each user.
- There is one master directory which contains separate directories dedicated to each user.
- For each user, there is a different directory present at the second level, containing group of user's file.
- The system doesn't let a user to enter in the other user's directory without permission.

Characteristics:

- Each file has a path name as /User-name/directory-name/
- Different users can have the same file name.
- Searching becomes more efficient as only one user's list needs to be traversed.

Advantages:

- Solve name collision problem as every user has separate directory.
- Independent user gets isolated from each others.

③ Hierarchical Directory System

- It is quite general and advantageous for users to group their files together in logical ways.

Advantage:

- With this directory system, in addition to their files, users can access the files of other users by specifying its pathname.

Path names:

- ① Absolute path name
- ② Relative path name

Directory Operations:

① Create

- Empty directory is created excluding dot and dotdot (. and ..) which are placed there automatically by the OS.

② Delete

- If only single dot and double dot (. and ..) is present in directory then it is treated as empty.
- Single dot and double dot within directory cannot usually be deleted.

③ Openness

- Directory can be read to list all the files from it.
- Directory should be open before reading just opening and reading the file.

④ Closedir

- After reading completes, a directory should be closed to free up inner table space.

⑤ Readdir

- It returns the next entry in an open directory. Readdir always returns one entry in a standard format irrespective of possible directory structure is being used.

⑥ Rename

- Directories can be renamed. Just like files, rename renames the directory.

⑦ Link

- Due to it, a file appears in more than one directory.

⑧ Unlink

- Unlink removes the directory entry.

File sharing

- Necessary to share a file among multiple users in multiuser system.
- Needs to deal with 2 issues

- ① Access Rights
- ② Simultaneous Access

Access Rights

① None

- The user would not be allowed to read the user directory that includes the file

② Knowledge

- The user can determine that the file exists and who its owner is and can then petition the owner for additional access right

③ Execution

- The user can load and execute a program but cannot copy it

④ Reading

- The user can read the file for any purpose, including copying and execution.

⑤ Appending

- The user can add data to the file but cannot modify or delete any of the file's content

⑥ Updating

- The user can modify, delete and add to the file's data

⑦ Changing protection

- The user can change the access rights granted to other users

⑧ Deletion

- The user can delete the file from the file system

Simultaneous Access

- When more than one 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 a collection of blocks
- The OS or file management system is responsible for allocating blocks to files
- The approach taken for file allocation may influence the approach taken for space management
- Space is allocated to a file as one or more portions.

Preallocation and Dynamic Allocation

- Preallocation policy requires that the maximum size of a file be declared at the time of the file creation request
- For many applications, it is difficult to estimate reliably the maximum potential size of the file.
 - Tends to be wasteful because users and application programmers tend to overestimate size.
- Dynamic allocates a space to a file portions as needed.

Portion size

- In choosing a portion size, there is a trade off between efficiency from the point of view of a single file v/s overall system efficiency

File Allocation Methods

① 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

② Linked List Allocation

- Each file carries list of links to disk blocks
- Directory contains link/pointer to first block of file
- No external fragmentation
- Effectively used in sequential access
- Inefficient in use of direct access

③ Indexed allocation

- provides solutions to problems of contiguous and linked allocation
- A index block is created having all pointers to files
- Each file has its own index block which stores the addresses of disk space occupied by the file
- Directory contains addresses of index blocks of files.

④ Linked list allocation using a table in memory

⑤ I-Nodes

- All types of UNIX files are administered by OS. by means of index nodes.
- An inode is a control structure that contains the key information needed by the OS for particular file

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

① 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 the block is empty then the bit is 1 otherwise it is 0.
- Initially all blocks are empty

② 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.
- Therefore, all the free blocks of the disk will be linked together with a pointer.
- Whenever a block gets allocated its previous free block will be linked to its next free block.

Linux Virtual File System

- ① 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 different types of file system.
- Linux of VFS is an interface between kernel and actual file system.
- ② The object oriented principles are used to design VFS.

- ③ 4 major object types by VFS -
 - An individual file is represented by i-node object.

- An open file is represented by file object
- An entire file system is represented as superblock object
- An individual directory entry is represented by entry object

- ④ A set of operations are defined for each of the type of objects. Each object of one of these types points to a function table.

- ⑤ The record of addresses of the actual functions is kept in function table.

- ⑥ The VFS software layer need not recognize earlier about what kind of object it is dealing with.