File Systems in Operating System

A file is a collection of related information that is recorded on secondary storage.

FILE DIRECTORIES:

A directory is a container that is used to contain folders and files. It organizes files and folders in a hierarchical manner.

Information contained in a device directory are:

- Name
- Type
- Address
- Current length
- Maximum length
- Date last accessed
- Date last updated
- Owner id
- Protection information

Operation performed on directory are:

- Search
- Create
- Delete
- List a directory
- Rename
- Traverse

Advantages of maintaining directories are:

- **Efficiency:** A file can be located more quickly.

- **Naming:** It becomes convenient for users as two users can have the same name for different files or may have different names for the same file.
- **Grouping:** Logical grouping of files can be done by properties e.g. all java programs, all games etc.

Two different methods of path name in a file directory are:-

1. Absolute Path name – In this method, each file is given an absolute path name consisting of the path from the root directory to the file. As an example, the path /usr/ast/mailbox means that the root directory contains a subdirectory usr, which in turn contains a subdirectory ast, which contains the file mailbox. Absolute path names always start at the root directory and are unique.

2. Relative Path name – A user can designate one directory as the current working directory, in which case all path names not beginning at the root directory are taken relative to the working directory.
   For example, if the current working directory is /usr/ast, then the file whose absolute path is /usr/ast/mailbox can be referenced simply as mailbox.

Absolute path name is used whenever programs need to access a specific file without regard to what the working directory is.

	DEFINITION	ADVANTAGES	DISADVANTAGES
Single-level directory	In this, all files are contained in the same directory which makes it easy to support and understand.	Easy to implement. Searching is faster in a small file. The operations like file creation, searching, deletion, updating are very easy in such a directory structure.	Chances of name collision increases. Searching takes time in large directories. Can't group the same type of files together.
Two-level directory	In this, each user has their own *user files directory (UFD)* which lists only the files of a single user. The system's *master file directory (MFD)* is searched whenever a new user is logged in. The MFD is indexed by username or account number, and each entry points to the UFD for that user.	Different users can have the same directory as well as the file name. Searching for files becomes easier due to pathname and user-grouping.	File sharing is not allowed. Two files of the same type can't be grouped together.
Tree-structured directory	It is the most common directory structure. The tree has a root directory, and every file in the system has a unique path. This generalization allows the user to create their own subdirectories and to organize their	Probability of name collision is very less. Searching becomes very easy.	Files may be saved into multiple directories. Sharing is not allowed. Inefficient.

	files accordingly.		
Acyclic graph directory	It allows us to share subdirectories and files. The same file or subdirectories may be in two different directories. The shared file is not the same as the copy file. If any programmer makes some changes in the subdirectory it will reflect in both subdirectories.	Files can be shared. Searching becomes easy.	Deletion may create problem due to sharing files.
General graph directory structure	In this, cycles are allowed within a directory structure where multiple directories can be derived from more than one parent directory.	Allow cycles More flexible.	More costly. Needs garbage collection.

Privileged Instructions

These are the instructions which can run only in kernel mode.

Various examples of Privileged Instructions include:

- I/O instructions and Halt instructions
- Turn off all Interrupts
- Set the Timer
- Context Switching
- Clear the Memory or Remove a process from the Memory
- Modify entries in the Device-status table

Non - Privileged Instructions

These are the instructions which can run only in user mode.

Various examples of Non - Privileged Instructions include:

- Reading the status of Processor
- Reading the System Time
- Generate any Trap Instruction
- Sending the final printout of Printer

FILE ALLOCATION METHODS

The allocation methods define how the files are stored in the disk blocks. There are three main disk space or file allocation methods.The main idea behind these methods is to provide efficient disk space utilization and fast access to the file blocks.

1. Contiguous Allocation

As the name suggests, each file occupies a contiguous set of blocks on the disk.
The file allocation table needs just a single entry for each file, showing the starting block and the length of the file. It is best from the point of view of the individual sequential file. Multiple blocks can be read in at a time to improve I/O performance for sequential processing. It is also easy to retrieve a single block.

Advantages:

- Both the Sequential and Direct Accesses are supported by this. For direct access, the address of the kth block of the file which starts at block b can easily be obtained as (b+k).
- This is extremely fast since the number of seeks are minimal because of contiguous allocation of file blocks.

Disadvantages:

- This method suffers from both internal and external fragmentation hence, makes it inefficient in terms of memory utilization.
- Increasing file size is difficult because it depends on the availability of contiguous memory at a particular instance.

2. Linked Allocation(Non-contiguous allocation)
In this scheme, each file is a linked list of disk blocks which can be scattered anywhere on the disk. The directory entry contains a pointer to the starting and the ending file block. Each block contains a pointer to the next block occupied by the file. Again the file table needs just a single entry for each file, showing the starting block and the length of the file. Any free block can be added to the chain.

Advantages:

- This is very flexible in terms of file size. File size can be increased easily since the system does not have to look for a contiguous chunk of memory.
- This method does not suffer from external fragmentation.

Disadvantage:

- Internal fragmentation exists in the last disk block of the file.
- There is an overhead of maintaining the pointer in every disk block.
- If the pointer of any disk block is lost, the file will be truncated.
- It supports only the sequential access of files.

3. Indexed Allocation

It addresses many of the problems of contiguous and chained allocation. In this case, the file allocation table contains a separate one-level index for each file: The index has one entry for each block allocated to the file. It supports both sequential and direct access to the file
We maintain a disk allocation table in addition to a file allocation table.The following are the approaches used for free space management.

3.1. Bit Tables :

A Bitmap or Bit Vector is a series or collection of bits where each bit corresponds to a disk block. The bit can take two values: 0 and 1 , 0 indicates that the block is allocated and 1 indicates a free block. It is simple to understand and finding first is efficient.

Block number :- *(number of bits per word) *(number of 0-values words) + offset of bit first bit 1 in the non-zero word .*

3.2. Linked List

In this approach, the free disk blocks are linked together i.e. a free block contains a pointer to the next free block. The block number of the very first disk block is stored at a separate location on disk and is also cached in memory. A drawback of this method is the I/O required for free space list traversal.

3.3. Grouping

This approach stores the address of the free blocks in the first free block. The first free block stores the address of some, say n free blocks. Out of these n blocks, the first n-1 blocks are actually free and the last block contains the address of the next free n blocks.
An **advantage** of this approach is that the addresses of a group of free disk blocks can be found easily.

3.4. Counting –This approach stores the address of the first free disk block and a number n of free contiguous disk blocks that follow the first block.
Every entry in the list would contain:

1. Address of first free disk block
2. A number n

Advantages:

- This supports direct access to the blocks occupied by the file and therefore provides fast access to the file blocks.
- It overcomes the problem of external fragmentation.

Disadvantages:

- The pointer overhead for indexed allocation is greater than linked allocation.

- For very small files, say files that expand only 2-3 blocks, the indexed allocation would keep one entire block (index block) for the pointers which is inefficient in terms of memory utilization. However, in linked allocation we lose the space of only 1 pointer per block.

File Access Methods

When a file is used, information is read and accessed into computer memory and there are several ways to access this information of the file.
There are three ways to access a file into a computer system: Sequential-Access, Direct Access, Index sequential Method.

1. Sequential Access

   It is the simplest access method. Information in the file is processed in order, one record after the other.

   Read and write make up the bulk of the operation on a file. A read operation *-read next-* read the next position of the file and automatically advance a file pointer, which keeps track of I/O location. Similarly, for the writer write *next* append to the end of the file and advance to the newly written material.

2. Direct Access

   It is also known as *relative access method*. A filed-length logical record that allows the program to read and write records rapidly in no particular order. The direct access is based on the disk model of a file for which the file is viewed as a numbered sequence of block or record. Thus, we may read block 14 then block 59 and then we can write block 17.

3. Index sequential method

   It is the other method of accessing a file which is built on the top of the sequential access method. These methods construct an index for the file which 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 a pointer we access the file directly.

Disk Scheduling Algorithms

It is done by operating systems to schedule I/O requests arriving for the disk. Disk scheduling is also known as I/O scheduling.
Terms related to Disk Scheduling Algorithms :

- **Seek Time :** It is the time taken to locate the disk arm to a specified track where the data is to be read or written.
- **Rotational Latency:** It is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads.
- **Transfer Time:** It is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred.

Disk Access Time = Seek Time + Rotational Latency + Transfer Time.

- **Disk Response Time:** It is the average of time spent by a request waiting to perform its I/O operation.

1. FCFS

It is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue.
Advantages:

- Every request gets a fair chance
- No indefinite postponement

Disadvantages:

- Does not try to optimize seek time

- May not provide the best possible service

2. SSTF

In SSTF (Shortest Seek Time First), requests having the shortest seek time are executed first. So, the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first.

Advantages:

- Average Response Time decreases
- Throughput increases

Disadvantages:

- Overhead to calculate seek time in advance
- Can cause Starvation for a request if it has higher seek time as compared to incoming requests
- High variance of response time as SSTF favours only some requests

3. SCAN

In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of the disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works as an elevator and hence also known as **elevator algorithm.** As a result, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait.

Advantages:

- High throughput
- Low variance of response time

- Average response time

Disadvantages:

- Long waiting time for requests for locations just visited by disk arm

4. CSCAN

In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.
So, instead of reversing its direction it goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to SCAN algorithm and hence it is known as C-SCAN (Circular SCAN).

Advantages:

- Provides more uniform wait time compared to SCAN

5. LOOK

It is similar to the SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

6. CLOOK

Similarly, CLOOK is also similar to CSCAN. In CLOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end's last request. Thus, it also prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

7. RSS

It stands for random scheduling and just like its name it is nature. It is used in situations where scheduling involves random attributes such as random processing time, random due dates, random weights, and stochastic machine breakdowns this algorithm sits perfect.

8. LIFO

In LIFO (Last In, First Out) algorithm, newest jobs are serviced before the existing ones i.e. in order of requests that get serviced the job that is latest or last entered is serviced first and then the rest in the same order.

Advantages

- Maximizes locality and resource utilization.

Disadvantages

- Can seem a little unfair to other requests and if new requests keep coming in, it causes starvation to the old and existing ones.

9. N-STEP SCAN

It is also known as the N-STEP LOOK algorithm. In this a buffer is created for N requests. All requests belonging to a buffer will be serviced in one go. Also once the buffer is full no new requests are kept in this buffer and are sent to another one. Now, when these N requests are serviced, the time comes for another top N requests and this way all get requests get a guaranteed service

Advantages

- It eliminates starvation of requests completely.

10. FSCAN

This algorithm uses two sub-queues. During the scan all requests in the first queue are serviced and the new incoming requests are added to the second queue. All new

requests are kept on halt until the existing requests in the first queue are serviced.

Each algorithm is unique in its own way. Overall Performance depends on the number and type of requests.

Introduction of Secondary Memory

Primary memory has limited storage capacity and is volatile. Secondary memory overcomes this limitation by providing permanent storage of data and in bulk quantity.

Fixed Storage-

A Fixed storage is an internal media device that is used by a computer system to store data, and usually these are referred to as the Fixed Disks drives or the Hard Drives.
Fixed storage devices are literally not fixed, obviously these can be removed from the system for repairing work, maintenance purpose, and also for upgrade etc.

Types of fixed storage:

- Internal flash memory (rare)
- SSD (solid-state disk) units
- Hard disk drives (HDD)

Removable Storage-

It is an external media device that is used by a computer system to store data, and usually these are referred to as the Removable Disks drives or the External Drives.
Examples of external devices include CDs, DVDs and Blu-Ray disk drives, as well as diskettes and USB drives.
This makes it easier for a user to transfer data from one computer system to another and provides the fast data transfer rates associated with storage area networks (SANs).

Types of Removable Storage:

- Optical discs (CDs, DVDs, Blu-ray discs)
- Memory cards
- Floppy disks
- Magnetic tapes
- Disk packs
- Paper storage (punched tapes , punched cards)

Secondary Storage Media

There are the following main types of storage media:

1. Magnetic storage media:

It is coated with a magnetic layer which is magnetized in clockwise or anticlockwise directions. When the disk moves, the head interprets the data stored at a specific location in binary 1s and 0s at reading.

Examples: hard disks, floppy disks and magnetic tapes.

- **Floppy Disk:** It is a flexible disk with a magnetic coating on it. It is packaged inside a protective plastic envelope. These are one of the oldest types of portable storage devices that could store up to 1.44 MB of data but now they are not used due to very less memory storage.
- **Hard disk:** It consists of one or more circular disks called platters which are mounted on a common spindle which is coated with a magnetic material. Both surfaces of each disk are capable of storing data except the top and bottom disk where only the inner surface is used. These heads are joined to a common arm known as access arm. Hard disks can store data upto several terabytes.

2. Optical storage media

In optical storage media information is stored and read using a laser beam. The data is stored as a spiral pattern of pits and ridges denoting binary 0 and binary 1.

Examples: CDs and DVDs

- **Compact Disk:** It is a device that a computer uses to read data that is encoded digitally on a CD and can be installed inside a computer's compartment, provided with an opening for easier disc tray access or it can be used by a peripheral device connected to one of the ports provided in the computer system. It can store approximately 650 to 700 megabytes of data.

- **DVD:**

  It stands for Digital Versatile Disk or Digital Video Disk. It looks just like a CD and uses a similar technology as that of the CDs but allows tracks to be spaced closely enough to store data that is more than six times the CD's capacity. A DVD holds 4.7 GB to 17 GB of data.

- **Blue Ray Disk:**

  This is the latest optical storage media to store high definition audio and video. It is similar to a CD or DVD but can store up to 27 GB of data on a single layer disk and up to 54 GB of data on a dual layer disk. While CDs or DVDs use a red laser beam, the blue ray disk uses a blue laser to read/write data on a disk.

3. Solid State Memories

Solid-state storage devices are based on electronic circuits with no moving parts like the reels of tape, spinning discs etc and use special memories called flash memory to store data. It is used mainly in digital cameras, pen drives or USB flash drives.

Pen Drives:

Pen Drives or Thumb drives or Flash drives are the recently emerged portable storage media. It is an EEPROM based flash memory which can be repeatedly erased and written using electric signals. This memory is accompanied with a USB connector which enables the pendrive to connect to the computer. They have a capacity smaller than a hard disk but greater than a CD.

Pendrive has following advantages:

- Transfer Files
- Portability
- Backup Storage
- Transport Data