UNIT 1: FILE SYSTEM

Unit Structure

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- Basic of File systems
- Structure and file partition concepts
- Types of files
- Non-contiguous and contiguous storage allocation
- Idea about Disk scheduling

1.1 Introduction

A file system is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk. The word is also used to refer to a partition or disk that is used to store the files or the type of the file system. Thus, one might say "I have two file systems" meaning one has two partitions on which one stores files, or that one is using the "extended file system", meaning the type of the file system.

The difference between a disk or partition and the file system it contains is important. A few programs (including, reasonably enough, programs that create file systems) operate directly on the raw sectors of a disk or partition; if there is an existing file system there it will be destroyed or seriously corrupted. Most programs operate on a file system, and therefore won't work on a partition that doesn't contain one (or that contains one of the wrong types).

Before a partition or disk can be used as a file system, it needs to be initialized, and the bookkeeping data structures need to be written to the disk. This process is called making a file system.

1.2 File Systems

When preparing a segment such as this, an adequate discussion tends to blur the line between hardware issues associated with hard disks and the software issues that control what is placed on them and in what manner. As manufacturers and operating system developers strive for performance and security, this line tends to blur even more. The very nature of the logical structures on a hard disk influences their performance, reliability, expandability and compatibility.

In spite of all of the media hype about them, a hard disk is merely a medium for storing information. A replacement for the limited capacity of the floppy disk, which was the first type of disk storage media available on small computers. As hard disks grow in capacity, becoming larger and larger every year, it is becoming increasingly difficult for operating systems and their companion file systems, to use them in an efficient manner.

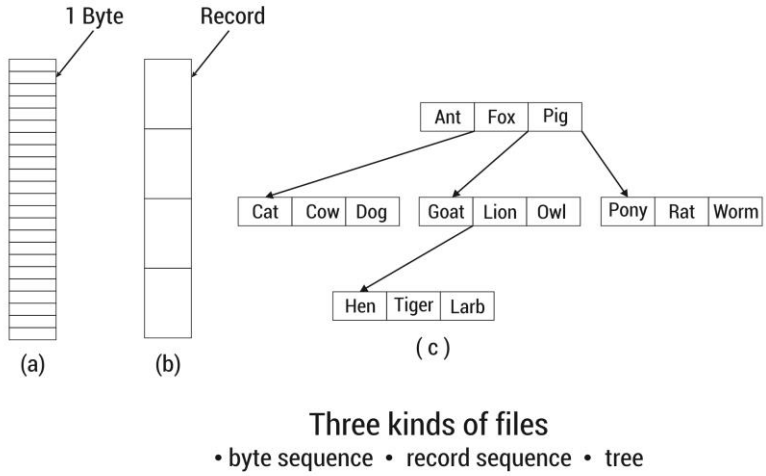

Fig 1.1 file system

The file system employed by most operating systems today is a generic name given to the software routines and logical structures used to prepare the given hard disk to store data as well as control access to that particular storage space. Different operating systems use different methods of organizing and controlling access to the data on the hard disk, which is entirely independent of the specific hardware in use. A single hard disk can be prepared in many different ways to store data, and under given circumstances a hard disk may even be prepared multiple ways on the same disk.

File systems will find about the naming particular files that are having maximum name characters to be utilised in certain systems and till what time the file name suffix can be applied. It shows a way to specify path to a file by the use of directory structure. It uses metadata to keep and retrieve files that will cover:

- Date created
- Date modified
- File size

Such type of file system example can be OS X that are utilised by Macintosh hardware by allowing various optimization features that will cover file names with 255 characters.

For certain group of user, such type of file system is constraints as they will not provide read / write access. The best way is to wither put a password or to encrypt the files so that the user can't access. While encrypting, a key is provided

to encrypt the file which can further decrypt the encrypted file text. The user with definite key can only access the required file.

1.2.1 Partitions

When referring to a computer hard drive, a disk partition or partition is a segment of the hard drive that is separated from other portions of the hard drive. Partitions help enable users to divide a computer hard drive into different drives or into different portions for multiple operating systems to run on the same drive.

With older file allocation tables, such as FAT16, creating smaller partitions allows a computer hard drive to run more efficiently and save more disk space. However, with new file allocation tables, such as FAT32, this is no longer the case.

There are different types of partitions that exist in Operating System:

AIX: Partition used with the AIX operating system.

Boot: It is a partition that contains the files required for a system start up.

BSD/OS: This partition is used with the BSD operating system.

DOS: It is used with older versions of MS-DOS.

DOS Ext: It is extended from one or more original MS-DOS partitions.

DRDOS: It is used with DR. DOS operating system.

Extended: It is extended from one or more of the primary partitions.

Hibernation: It is used with older hibernation programs.

HPFS: This is used with IBM OS/2 and Microsoft NT 3.x

Linux: This partition is used with several variants of Linux O/S.

MINIX: This is used with MINIX operating system.

NON-DOS: It is used in Microsoft fdisk partition which is not native to Microsoft operating system.

NEC DOS: It is used with earlier NEC DOS variant.

NEXTSTEP: It is used with Next step operating system.

Novell Netware: It is used with Novell Netware operating system.

NTFS: It is used with Microsoft Windows NT 4.x, Windows 2000 and Windows XP.

Partition Magic: It is created using Partition Magic utility by PowerQuest.

PC-ARMOUR: It is created by PC ARMOUR security utility.

Solaris X86: This is used with Sun Solaris X86 platform operating system.

System: This partition contains system32 directory.

Tandy DOS: It is used with earlier Tandy DOS variant.

Unix System V: This is used with various Unix Operating systems.

VMWare: This is used by VMWare.

XENIX: It is used with Xenix operating system.

1.2.2 Directory structure

There are many types of directory structure in Operating System. They are as follows:-

1) Single Level Directory
2) Two Level Directory
3) Tree Structured Directory
4) Acyclic Graph Directory
5) General Graph Directory

Single Level Directory

In this type of directory structure as shown in fig 1.2, all files are in the same directory.

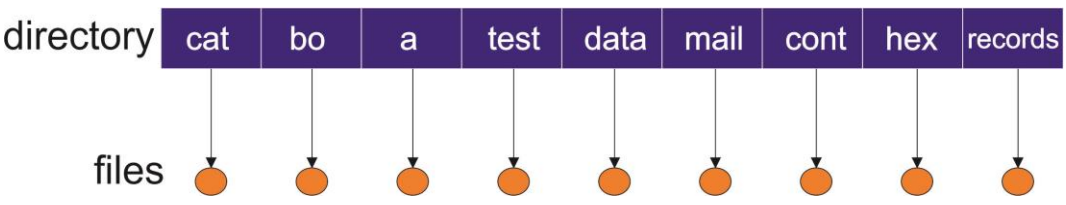

Fig 1.2 Single level Directory

It has certain limitations as:

- Being all files in same directory, they possess unique name.
- If two users call their data free test, then unique name rule is violated.
- Files are limited in length.
- Even a single user may find it difficult to remember the names of all files as the number of file increases.
- Keeping track of so many file is daunting task.

Two Level Directories

In this type of directory system as shown in fig 1.3:

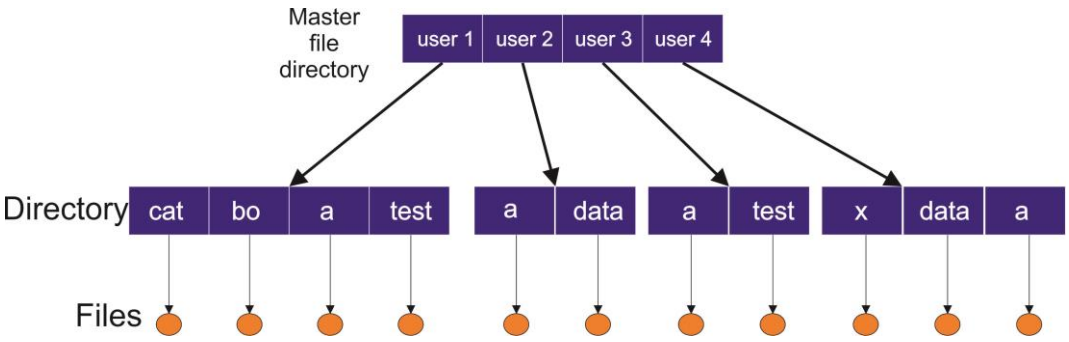

Fig 1.3 Two level Directory Structure

i. Every user has its own User File Directory (UFD).

ii. On account of user job start or log in, system Master File Directory (MFD) gets searched. The Master File Directory gets indexed by user name or Account Number.

iii. In case when the user refers to certain file, then at that time its own UFD gets searched.

It is seen that different users contains files in similar name. To obtain a different partition, in a two level directory, we assign user name and file name. In case of two level directories, we see that it could be a tree or an inverted tree of height 2. Also the root of a tree in this is a Master File Directory (MFD) which gets direct descendants as User File Directory (UFD). Descendants of UFD's are its file. Such files are the leaves of a tree. This type of file structure has certain limitations as the structure become isolated from one user to another.

Tree Structured Directory

It is another type of directory or Sub directory structure as shown in fig 1.4 carries set of files or sub directories.

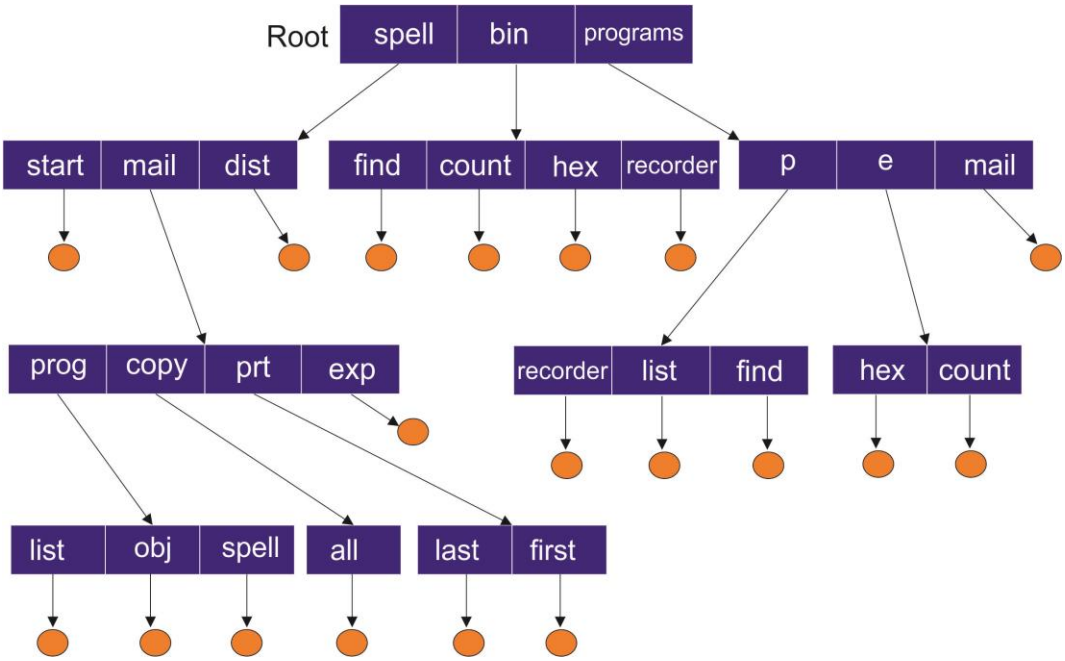

Fig 1.4 Tree Level Directory Structure

In this, complete directories contains similar internal format. It has specific features:

1. In this, single bit in each directory entry shows the entry.
2. Here the special calls are used to form and delete directories.
3. In this all process has current directory that carries many files which are of current interest to the process.
4. In this, when a reference is made to a file, the current directory gets searched.
5. In this, the user will amend his current directory whenever he wants.
6. In case, if the file is not required in present directory then the user normally either shows a path name or practically amends the present directory. Here the paths can be of two types:-
   a) Absolute Path: This will start at root and follows a path down to particular file.
   b) Relative Path: this will explain a path from present directory.
7. In this type of directory structure, if deleted directory is empty, then its entry in directory containing will get deleted. On the other hand, if the directory is not empty, then either the two approaches exist:-
   a) User must delete all the files in the directory.

b) If any sub directories exist, same procedure must be applied.

Here the UNIX rm command is used, whereas MS dos will not delete a directory unless it is empty.

Acyclic Graph Directory

Fig 1.5 shows another type of directory which is a graph having no cycles.

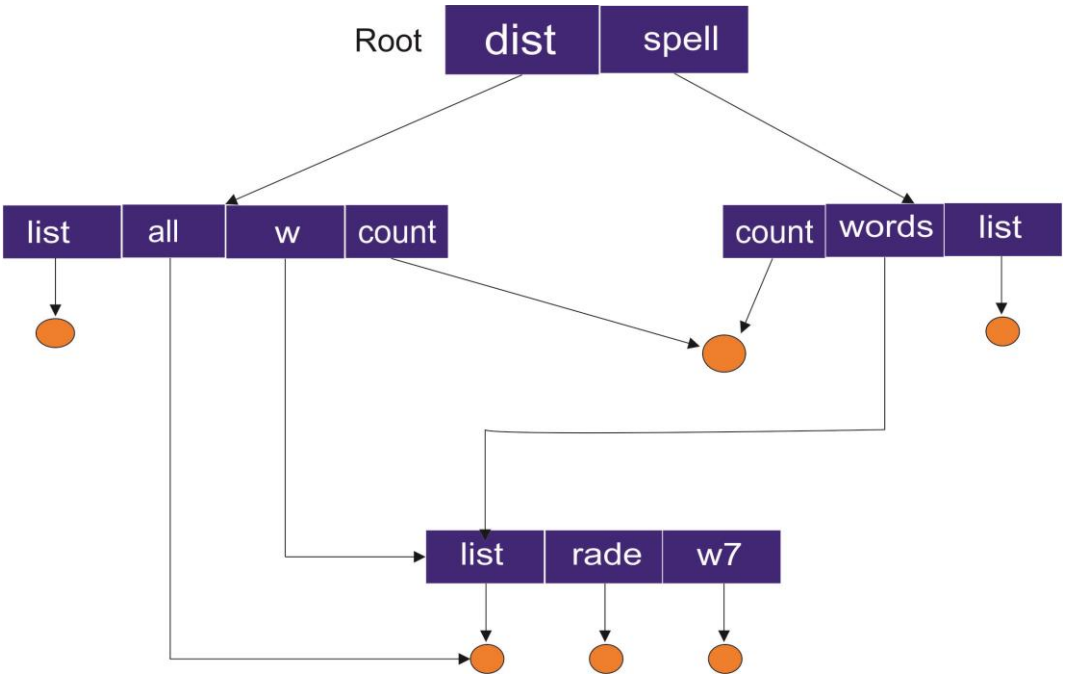

Fig 1.5 Acyclic Graph Directories

This type of directory structure allows directories to contribute sub directories and files. In sharing mode, only actual file exists, so for any changes by one person gives visibility to another. Here the implementation of shared files and directories takes place:

A. Creating a link

- A link is effectively a pointer to another file or sub directory.
- Duplicate all the information about them in both sharing directories.

B. Deleting a link

- Deletion of the link will not affect the original file, only link is removed.
- To preserve the file until all references to it are deleted.

Check your progress 1

1. Which is not a file type?

   a. tree c. byte sequence

   b. leaf sequence d. record sequence

2. Which is an example of metadata tags?

   a. Date created c. File size

   b. Date modified d. all

3. Which partition is used with older versions of MS-DOS.

   a. DOS c. DRDOS

   b. DOS Ext d. none

4. In which type of directory arrangement, all files are placed in single directory.

   a. Single Level Directory c. Tree Structured Directory

   b. Two Level Directory d. Acyclic Graph Directory

1.3 Disk Space Allocation

It is noted that an important function of a file system is to handle the space present on the secondary storage with the use of safe tracking of disk blocks which are assigned to files along with free blocks which is made available for allocation. The process of allocation of space to files carries the following problems:

1. Efficient disk space usage
2. Quick file access

Disk block management appears to be a problem where the secondary storage arises with two additional problems as:

1. Low disk access time
2. Dealing of more blocks

With such problems, there appears to be plenty of conclusions which is present in both environments such as contiguous file allocation as well as non-contiguous file allocation. With this, there exists three another allocation

techniques like contiguous, linked along with indexed. It is finding that every method has its own merits and demerits.

Disk Allocation Methods

By using direct applications of disks and storing of files in neighbouring part the disk is recommended. The only problem arises as how to give space to files in order to have efficient disk space application with faster response. Once the files are allotted and not tied up, the space available on the disk gets splitted up. There are two important methods, where the disk space gets allotted which are:

1. Continuous
2. Non-continuous

Contiguous Allocation

The contiguous allocation involves assigning files to contiguous secondary storage space. For this, the programmer or user needs to present in advance the size of area that can keep a file that is to be framed. If the particular portion of the space is not free, then the file will not be formed.

The benefit in case of contiguous allocation is that nearly every records of a file will appear closely to each other directly which will increases the entry speed of particular records. This will explain that, if the records are placed here and there across the disk, then its accessing speed will be slow. For a contiguously allocated file, the addressing is easy. In case of sequential access, the file system will keep disk address of last block and if required will reads the next block. In the diagram shown in fig 1.6, in order to access directly to block B of a file having location L, it is defined that it will quickly shows the block L+B. So with this, we can say that contiguous allocation is applicable to both sequential as well as direct accessing.

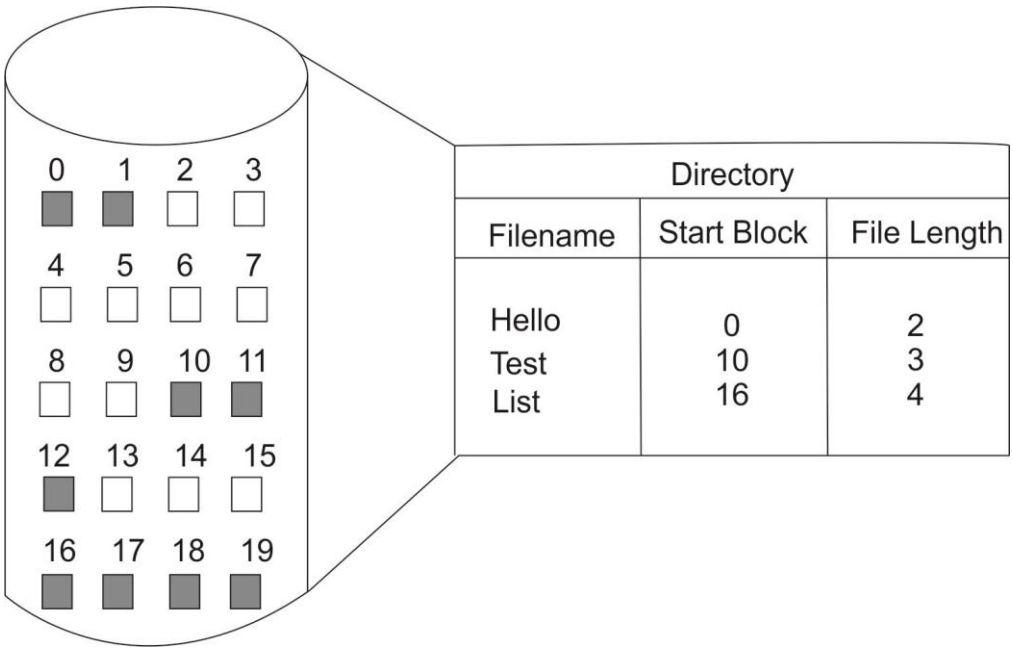

Fig 1.6 contiguous allocation systems

Also it is noted that the file directories available inside contiguous allocation systems is clear to implement. It is found that every file is required to keep the initial location of particular file along with required file size. Consider again the diagram shown in fig 1.6, where the file size is of N blocks long which originates from location L, then it will gather blocks L, L+ 1 .....L+N- 1. In this the directory entry shows the location or address of initial block along with the file size length.

The only problem with in the contiguous allocation is locate for the space inside a new file when process of free space listing is performed with the help of bit map method. In order to create n-bytes which are quiet long file, we should first locate for n 0 bits in row. For better understanding of contiguous storage allocation problem, we need to assume disk space as a mixture of free as well as occupied portion. With this, we feel that each segment is contiguous group of disk blocks.

In order to agree a request of file for n free contiguous blocks, a group of unused blocks can be find out which is find and decide that which hole will be best for allocation. Such type of problem refers to particular application of dynamic storage allocation which will explain how a request is to satisfied with N size request from a particular list that carries free holes. For this you need to follow two common strategies which can be

1. First – fit

In case of first fit strategy as soon as first hole appears, you will see that the searching gets paused moreover the memory is billed for developing a file. Here, searching will start at the start of set of holes or from the place where earlier first fit search gets halted.

2. Best – fit

In case of best fit strategy, the searching of complete list started for smallest hole, which is quiet big that builds for developing a particular file.

From the above two strategies, we see that both these are not fit in case of storage applications. It is seen that out of the two strategies, first-fit is normally faster as compared to best fit.

With this reason, such type of algorithms will lack from external fragmentations. Once the files are billed and erased, the empty space on the disk space gets broken down into smaller pieces. So we can say that external fragmentations is basically scattered groups that has free blocks which are very tiny for allocation that on collection will show big disk size.

As per the full disk storage and related file size, it seems that external fragmentation could result in minor or major problem. If we see the major problem in case of contiguous allocation, we will find that it is difficult to find the space which is required by the file system. Such type of problems will not arise in case of copying files where exact determination of file size is hard and is not correct.

When the expected file size is much larger in a way that its extension is kept in different disk area, then such of mechanism is called as file overflow. It is seen that locating overflowed contiguous area is quiet boring and difficult that will lost a feel in regard of contiguous allocation.

Non Contiguous Allocation

As the user are unaware about the file storage capacity and data present to contiguous , so nowadays, the storage allocation systems are changed with more dynamic non-contiguous storage allocation systems, that can be:

- Linked allocation
- Indexed allocation

Linked Allocation

It is seen that linked allocation is normally a disk based description of linked list, where the disk blocks are placed here and there on the disk. In this, a directory consists of pointer which is placed in first and last block of file. Also each block contains pointers to the next block, which are not made available to the user.

It can be used effectively for sequential access only but there also it may generate long seeks between blocks. Another issue is the extra storage space required for pointers. Yet the reliability problem is also there due to loss/damage of any pointer. The diagram 1.7 shows linked /chained allocation where each block contains the information about the next block.

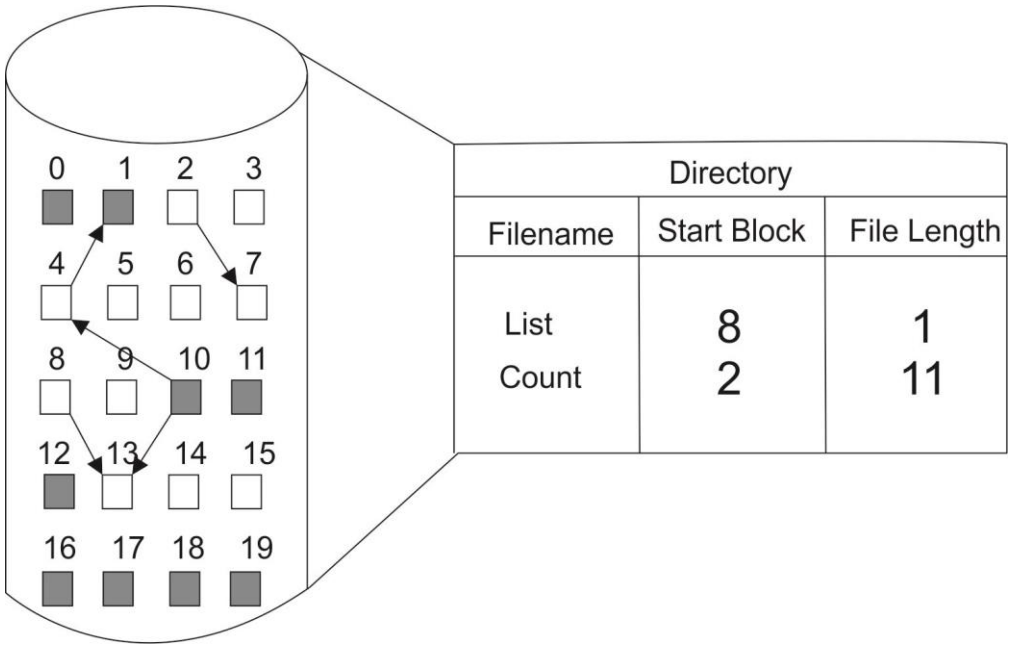

Fig 1.7 linked /chained allocation

MS-DOS and OS/2 use another variation on linked list called FAT (File Allocation Table). The beginning of each partition contains a table having one entry for each disk block and is indexed by the block number.

The directory entry contains the block number of the first block of the file. The table entry indexed by block number contains the block number of the next block in the file.

The Table pointer of the last block in the file has an EOF pointer value. This chain continues until EOF (end of file) table entry is encountered.

With this, we will pass to the next pointers without entering inside the disk for all. The 0 table value shows the presence of unused block; hence allocation of

free blocks with the help of FAT arrangement is clear with simple searching of first block along with 0 table pointer. MS-DOS and OS/2 use this scheme. The figure 1.8 shows file allocation table (FAT).

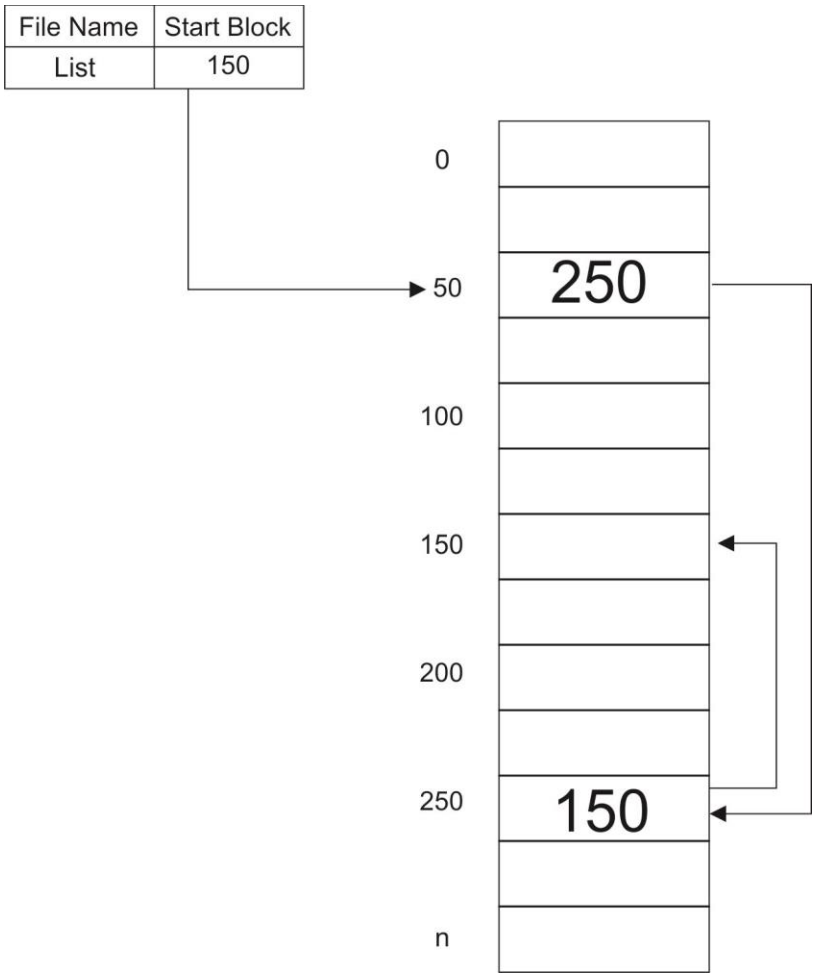

Fig 1.8 Directories

Indexed Allocation:

Index allocation 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 portion allocated to file.

Typically, the file indexes are not physically stored as part of the file allocation table. Rather, the index for a file is kept in a separate block, and entry for the file in the allocation table points to that block. The allocation may be on the basis of either fixed size blocks or variable size portions. The indexed allocation scheme is shown in fig 1.9.

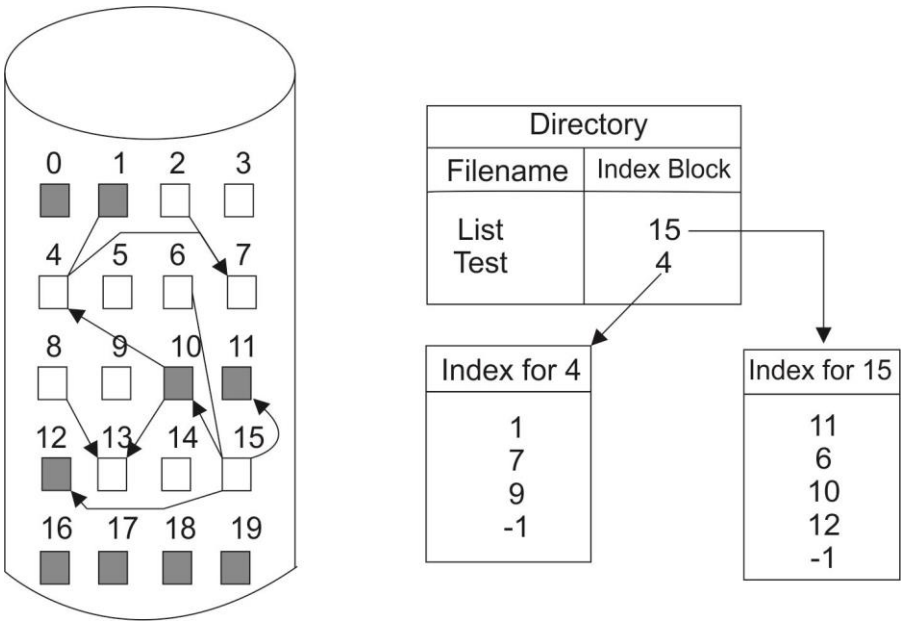

Fig 1.9 indexed allocation

The advantage of this scheme is that it supports both sequential and random access. The searching may take place in index blocks themselves. The indexes blocks may be kept close together in secondary storage to minimize seek time. Also space is wasted only on the index which is not very large and there's no external fragmentation.

Check your progress 2

1. Which is a hindrance of secondary storage?

   a. slows disk access time

   b. larger number of blocks to deal with

   c. both a and b

   d. neither a nor b

2. Which is incorrect in case of Contiguous Allocation?

   a. files are assigned to contiguous areas of secondary storage

   b. specific size by the user

   c. file can be created with any space requirement

   d. successive records of file are adjacent to each other

1.4 Disk Scheduling

It is seen that in an individual Computer, there can be many operations at a particular time; hence the management is required on all running processes that are running on particular system at a time. The idea of multi-programming is to perform multiple programs at the same time. To control and distribute memory to related processes, operating system will try to utilise the disk scheduling procedure.

With this process, the CPU time is distributed among various related processes which will describe all procedures in order to perform well. These processes will be specified by disk scheduling that specifies which process to be executed initially by the CPU. Normally, scheduling is concerned with processes that are taken care by the CPU during the particular time.

The idea of CPU scheduling is to examine the complete time required by CPU which carries certain number or processes which can occur simultaneously during a particular period of time. In order to share or divide the complete time of CPU, the CPU will make use of following scheduling processes:

FCFC or First Come First Serve -

In First Come First Serve process, Job or Processes which are undertaken gets arranged as per their order to entry inside the Computer system. The role of Operating System which is located in a queue has series of order which will be acted and describe for the whole process. In this, the jobs are carried out in the same manner as they entered inside the computer system.

SSTF or Shortest Seek Time First -

This type of technique is basically employed by an operating system in order to find for the minimum time. With this technique, searching will takes less time in order to find for a job. After the final examination, all jobs gets arranged in certain sequence as per the priority. The priority in this will signify the total processing time required by particular job to work out. In this the shortest seek time will encounter all time which will takes up the time that to be entered and completes the process.

C-Scan Scheduling -

In such type of scheduling, the processes get arranged by using particular circular order list. Circular List contains a process having no start and end point. In this, the end of List will actually know as the starting point of list. This scheduling involves CPU searching from start to end of a particular process and if

an end was taken, then it will again start its process from starting process. This happens as many times, when a CPU is working on a process, and then the user may wish to enter some data. It shows that the user if required can enter some data as the situation arises where the CPU will again execute the process soon after the Input operation. This type of scheduling is mostly applied in order to process same process in a cycle.

Look Scheduling -

This scheduling involves complete CPU scanning of list from start to end of Disk along with certain other procedures. This scheduling involves continuous CPU scanning of complete disk again and again from one point end to another end point.

Round Robin -

This scheduling is distributed proportionately which is known as Quantum Time. In this, each process that ask for execution gets similar amount of CPU times with standard quantum time. In this the initial process is present where CPU enters straight into next process state. Such type of scheduling is not favourable where after completion of process; the time also gets used by the process. This will shows that due to the presence of dissimilar operation, CPU time gets absorbed by CPU itself which further is wastage.

Priority Scheduling -

In this scheduling, every prioritized process gets checked with the help of total time which is carried out by such processes. This scheduling will examine the complete process time along with total number of Input/output process so that to stop priorities of processes.

Multilevel Queue:

This scheduling is applicable when there are many queues for definite number of processes. This happens as we know that there are many works which are to be performed with computers during the particular time. To arrange different Queues, the CPU here will arrange such Queues by using certain type of approach. The queues will assemble and are arranged in definite functions whose request is there to work.

Check your progress 3

1. Which technique is used by the Operating System to search for the shortest time?

   a. FCFC c. C-Scan

   b. SSTS d. Look Scheduling

2. In which scheduling, the processes get arranged by using particular circular order list.

   a. FCFC c. C-Scan

   b. SSTS d. Look Scheduling

3. In case of _____scheduling, the time of CPU is shared among equal numbers as Quantum Time.

   a. Round Robin c. C-Scan

   b. Look d. none

1.5 Let Us Sum Up

In this unit, we have learned:

- MS-DOS and OS/2 use another variation on linked list called FAT.
- Index allocation addresses many of the problems of contiguous and chained allocation.
- C-Scan Scheduling is a type of scheduling, where the processes get arranged by using particular circular order list.
- Round Robin is a type of scheduling where the time of CPU is shared into equal numbers which is called as Quantum Time.

1.6 Answers for Check Your Progress

Check your progress 1

Answers: (1-b), (2-d), (3-a), (4-a)

Check your progress 2

Answers: (1-d), (2-c)

Check your progress 3

Answers: (1-b), (2-c), (3-a)

1.7 Glossary

1. **File -** A file is a collection of records.
2. **File Organisation -** It is way by which the records get accessed on the disk.
3. **Sequential File -** It is a simplest file organisation where sequential access is present instead of individual file.
4. **Index allocation -** Type of file system that addresses many problems of contiguous and chained allocation.
5. **C-Scan Scheduling -** A scheduling where the processes get arranged by using particular circular order list.
6. **Round Robin -** It is a scheduling where the time of CPU is shared into equal numbers which is called as Quantum Time.

1.8 Assignment

Explain the Operating System File structure?

1.9 Activities

Study file organisation in Operating System.

1.10 Case Study

Study the types of partition in Windows Operating System.

1.11 Further Readings

1. Operating System Concept by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne.
2. Programming Be Operating System by Dan Sydow.

3. Computer Science & Application by Dr. Arvind Mohan Parashar, Chandresh Shah, Saurab Mishra.

4. An Integrated Approach to Software Engineering by Pankaj Jalote.

5. An Operating Systems by Raphael A.