

# FILE ALLOCATION

The allocation method defines how the files are stored in the disk blocks. The direct access nature of the disks gives us the flexibility to implement the files. In many cases, different files or many files are stored on the same disk. The main problem that occurs in the operating system is that how we allocate the spaces to these files so that the utilization of disk is efficient and the quick access to the file is possible. There are mainly three methods of file allocation in the disk. Each method has its advantages and disadvantages. Mainly a system uses one method for all files within the system.

- Contiguous allocation
- Linked allocation
- Indexed allocation

The main idea behind contiguous allocation methods is to provide :

- Efficient disk space utilization
- Fast access to the file blocks

## Contiguous File Allocation :

In this scheme, a file is made from the contiguous set of blocks on the disk. Linear ordering on the disk is defined by the disk addresses. In this scheme only one job is accessing the disk block  $b$  after that it accesses the block  $b+1$  and there are no head movements. When the movement of the head is needed the head moves only from one track to another track. So the disk number that is required for accessing the contiguous allocation is minimal. Contiguous allocation method provides a good performance that's why it is used by the IBM VM/CMS operating system. 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:  $b, b+1, b+2, \dots, b+n-1$ . 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. For a contiguous allocation the directory entry the address of the starting block and Length of the allocated portion.

### Advantages:

1. In the contiguous allocation, sequential and direct access both are supported.
2. For the direct access, the starting address of the  $k$ th block is given and further blocks are obtained by  $b+K$ ,
3. This is very fast and the number of seeks is minimal in the contiguous allocation method.

### Disadvantages:

1. Contiguous allocation method suffers internal as well as external fragmentation.
2. In terms of memory utilization, this method is inefficient.

3.It is difficult to increase the file size because it depends on the availability of contiguous memory.

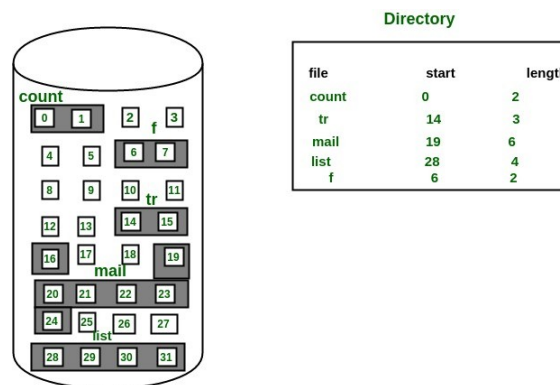


FIG 1. Contiguous file allocation on the disk

## Linked File Allocation :

The problems of contiguous allocation are solved in the linked allocation method. In this scheme, disk blocks are arranged in the linked list form which is not contiguous. The disk block is scattered in the disk. In this scheme, the directory entry contains the pointer of the first block and pointer of the ending block. These pointers are not for the users. For example, a file of six blocks starts at block 10 and end at the block. Each pointer contains the address of the next block. When we create a new file we simply create a new entry with the linked allocation. Each directory contains the pointer to the first disk block of the file. when the pointer is nil then it defines the empty file.

### Advantages:

- 1.In terms of the file size, this scheme is very flexible.
- 2.We can easily increase or decrease the file size and system does not worry about the contiguous chunks of memory.
- 3.This method free from external fragmentation this makes it better in terms of memory utilization.

### Disadvantages:

- 1.In this scheme, there is large no of seeks because the file blocks are randomly distributed on disk.
- 2.Linked allocation is comparatively slower than contiguous allocation.
- 3.Random or direct access is not supported by this scheme we cannot access the blocks directly.
- 4.The pointer is extra overhead on the system due to the linked list.

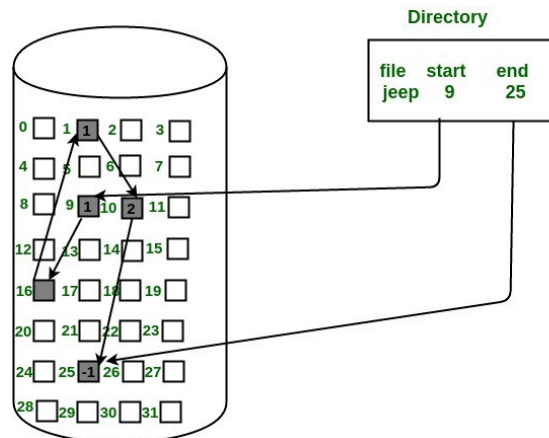


FIG 2. Linked file allocation on the disk.

## Indexed File Allocation :

In this scheme, a special block known as the index block contains the pointer to all the blocks occupied by a file. each file contains its index which is in the form of an array of disk block addresses. The  $i$ th entry of index block point to the  $i$ th block of the file. The address of the index block is maintained by the directory. When we create a file all pointer is set to nil. A block is obtained from the free space manager when the first  $i$ th block is written. When the index block is very small it is difficult to hold all the pointers for the large file. to deal with this issue a mechanism is available. Mechanism includes the following:

- Linked scheme
- Multilevel scheme
- Combined scheme

### Advantages:

- 1.This scheme supports random access of the file.
- 2.This scheme provides fast access to the file blocks.
- 3.This scheme is free from the problem of external fragmentation.

### Disadvantages:

- 1.The pointer head is relatively greater than the linked allocation of the file.
- 2.Indexed allocation suffers from the wasted space.
- 3.For the large size file, it is very difficult for single index block to hold all the pointers.
- 4.For very small files say files that expend only 2-3 blocks the indexed allocation would keep on the entire block for the pointers which is insufficient in terms of memory utilization.

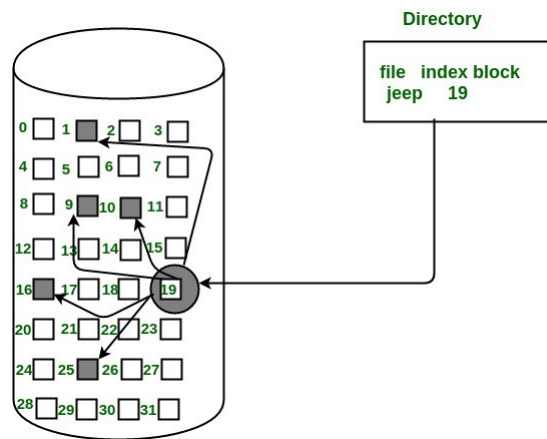


FIG 3. Indexed File Allocation