**Question Bank**

**2 Marks Questions**

1. **What is a File?**

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks

1. **List the various File Attributes.**
   * 1. Name
     2. Identifier
     3. Type
     4. Location
     5. Size
     6. Protection
     7. Time, data and User Identification
2. **What are the various File Operations?**
   * + - 1. Creating a file
         2. Writing a file
         3. Reading a file
         4. Delete a file
         5. Truncating the file
         6. Repositioning within a file
3. **What are the different Accessing Methods of a File?**
   * + - 1. Sequential Access
         2. Direct Access
4. **What is Directory?**

A **directory** is a location for storing files on your computer. **Directories** are found in a hierarchical file system, such as Linux, MS-DOS, and Unix.

1. **What do you mean by relative path**

Relative path is defined as the path related to the present working directory.

1. **Explain absolute path name**

An absolute path refers to the complete details needed to locate a file or folder, starting from the root element and ending with the other subdirectories.

Example: Windows **\usr\ast\mailbox**

1. **What are the most common schemes for defining the Logical Structure of a Directory?**

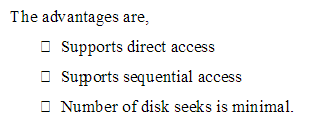
Single-Level-Directory  
Two-level-Directory  
Tree-Structured-Directory  
Acyclic-Graph-Directory  
 General Graph-Directory

1. **What is a Path Name?**

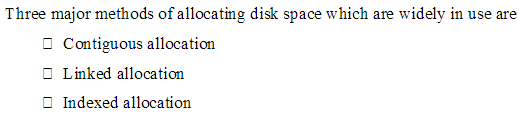
A pathname is the path from the root through all subdirectories to a specified file.

In a two level directory structure a user name and a file name define a path name.

1. **What are the advantages of Contiguous Allocation?**



1. **What are the types of path name**
   1. Relative Path Name
   2. Absolute Path Name
2. **What are the Allocation Methods of a disk space**



1. **What do you mean by sequential access method**

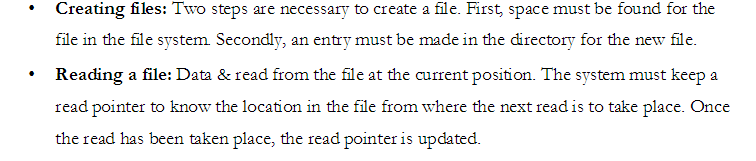
Sequential access is the [reading](https://www.webopedia.com/TERM/R/read.html) or [writing](https://www.webopedia.com/TERM/W/write.html) of [data](https://www.webopedia.com/TERM/D/data.html) [records](https://www.webopedia.com/TERM/R/record.html) in sequential order, that is, one record after the other. To read record 10, for example, you would first need to read records 1 through 9.

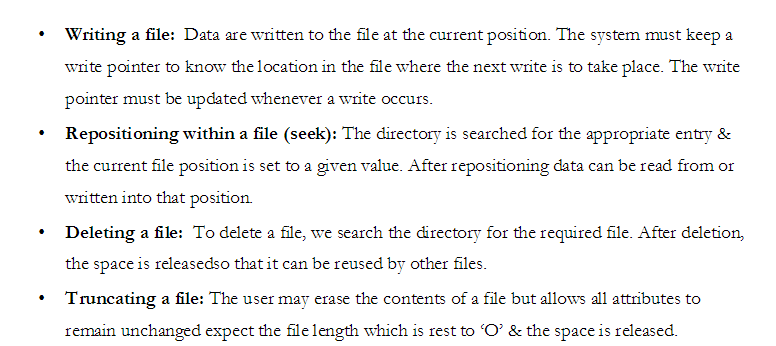
1. **Define direct access method in files.**

A method of reading data from a computer file without reading through the file from the beginning as on a disk. If you want to access particular in information in the disk you can access directly only those information.

**5 Marks Questions**

1. **List and explain the various File Attributes.**
2. **Name.** It is the only information which is in human-readable form.
3. **Identifier**. The file is identified by a unique tag(number) within file system.
4. **Type**. It is needed for systems that support different types of files.
5. **Location**. Pointer to file location on device.
6. **Size**. The current size of the file.
7. **Protection**. This controls and assigns the power of reading, writing, executing.
8. **Time, date, and user identification**. This is the data for protection, security, and usage monitoring.
9. **Explain various file operations**

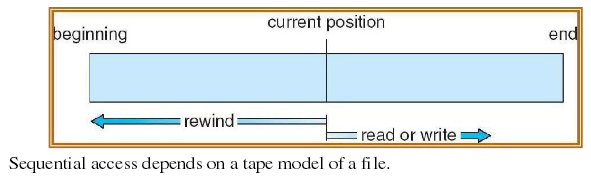




1. **Explain different file accessing method**

**Sequential Access:**

1. Sequential access is the simplest access method. Information in the file is processed in order, one record after another.
2. Editors and compilers access the files in this fashion.
3. Normally read and write operations are done on the files.
4. A read operation reads the next portion of the file and automatically advances a file pointer, which track next i/I track.
5. Write operation appends to the end of the file and such a file can be next to the beginning.

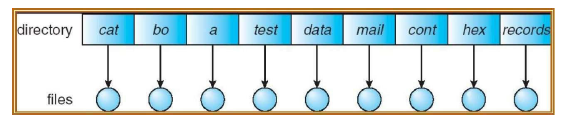


**Direct Access**

1. Direct access allows random access to any file block.
2. This method is based on disk model of a file.A file is made up of fixed length logical records. It allows the program to read and write records rapidly in any order.A direct access file allows arbitrary blocks to be read or written.
3. **Eg**:-User may need block 13, then read block 99 then write block 12. For searching the records in large amount of information with immediate result, the direct access method is suitable. Not all OS support sequential and direct access. Few OS use sequential access and some OS uses direct access.
4. It is easy to simulate sequential access on a direct access but the reverse is extremely inefficient.
5. **Explain the various Directory structures**

**1. Single level directory:** It is the simplest directory structure. All files are present in the same Directory. So it is easy to manage & understand.

**Limitation:** A single level directory is difficult to manage when the no. Of files increases or When there is more than one user. Since all files are in same directory, they must have unique Names. So, there is confusion of file names between different users.

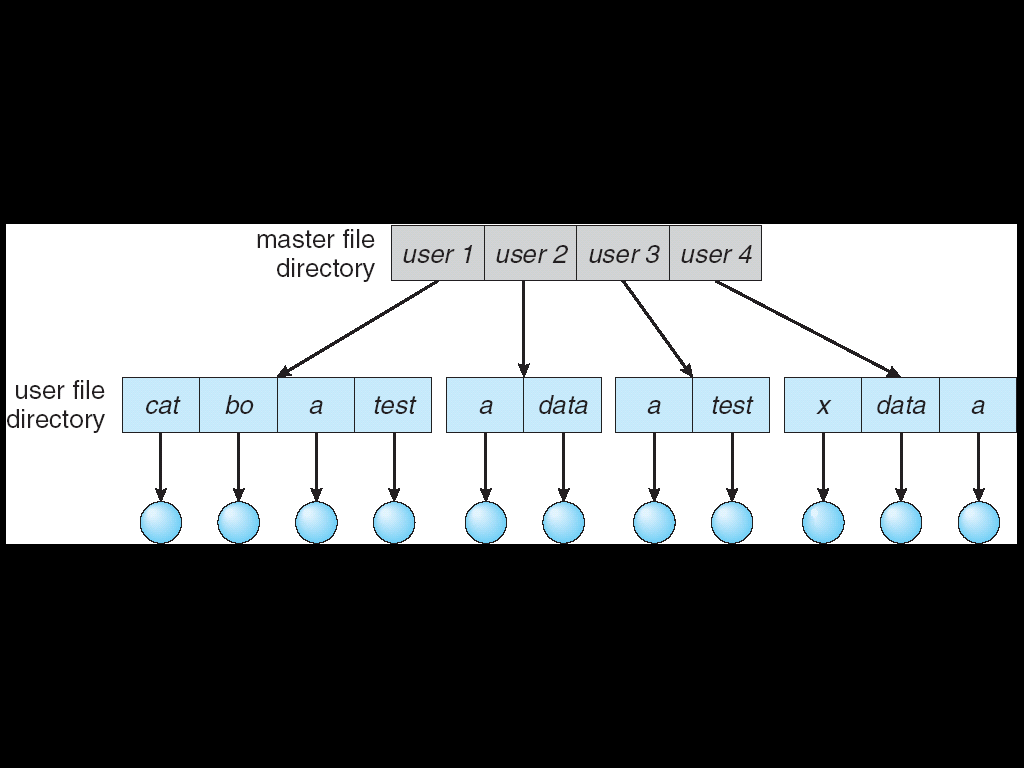


1. **Two Level Directory:**

The solution to the name collision problem in single level directory is to

Create a separate directory for each user. In a two level directory structure, each user has its own user file directory. When a user logs in, then master file directory is searched. It is indexed by user name & each entry points to the UFD of that user.

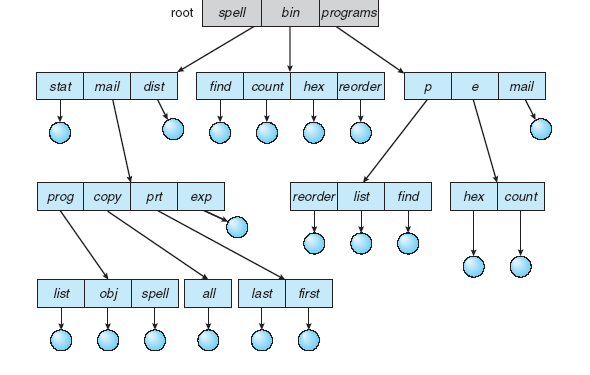
**Limitation:** It solves name collision problem. But it isolates one user from another. It is an advantage when users are completely independent. But it is a disadvantage when the users need



1. **Tree structured directories:**

* It is the most common directory structure. A two level directory is a two level tree. So, the generalization is to extend the directory structure to a tree of arbitrary height.
* It allows users to create their own subdirectories & organize their files. Every file in the system has a unique path name. It is the path from the root through all the sub-directories to a specified file.
* A directory is simply another file but it is treated in a special way. One bit in each directory entry defines the entry as a file (O) or as sub- directories. Each user has a current directory.
* It contains most of the files that are of current interest to the user. Path names can be of two types: An absolute path name begins from the root directory & follows the path down to the specified files. A relative path name defines the path from the current directory.

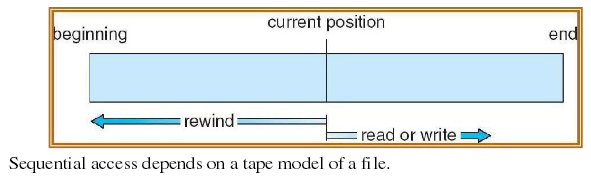
**E.g.** If the current directory is root/spell/mail, then the relative path name is prt/first & the absolute pathname is root/ spell/ mail/ prt/ first. Here users can access the files of other users also by specifying their path names.



1. **With diagram explain how to access file with sequential access**

**Sequential Access:**

1. Sequential access is the simplest access method. Information in the file is processed in order, one record after another.
2. Editors and compilers access the files in this fashion.
3. Normally read and write operations are done on the files.
4. A read operation reads the next portion of the file and automatically advances a file pointer, which track next i/I track.
5. Write operation appends to the end of the file and such a file can be next to the beginning.



1. **List and explain what all the operations performed on directory.**

* **Search for a file:** We need to be able to search a directory for a particular file.
* **Create a file:** New files are created & added to the directory.
* **Delete a file:** When a file is no longer needed, we may remove it from the directory.
* **List a directory:** We should be able to list the files of the directory.
* **Rename a file:** The name of a file is changed when the contents of the file changes.
* **Traverse the file system:** It is useful to be able to access every directory & every file within a directory.

**7. With a diagram explain file system structure**

To provide efficient and convenient access to the disks, the OS provides the file system to allow the data to be stored, located and retrieved. A file system has two design problems:

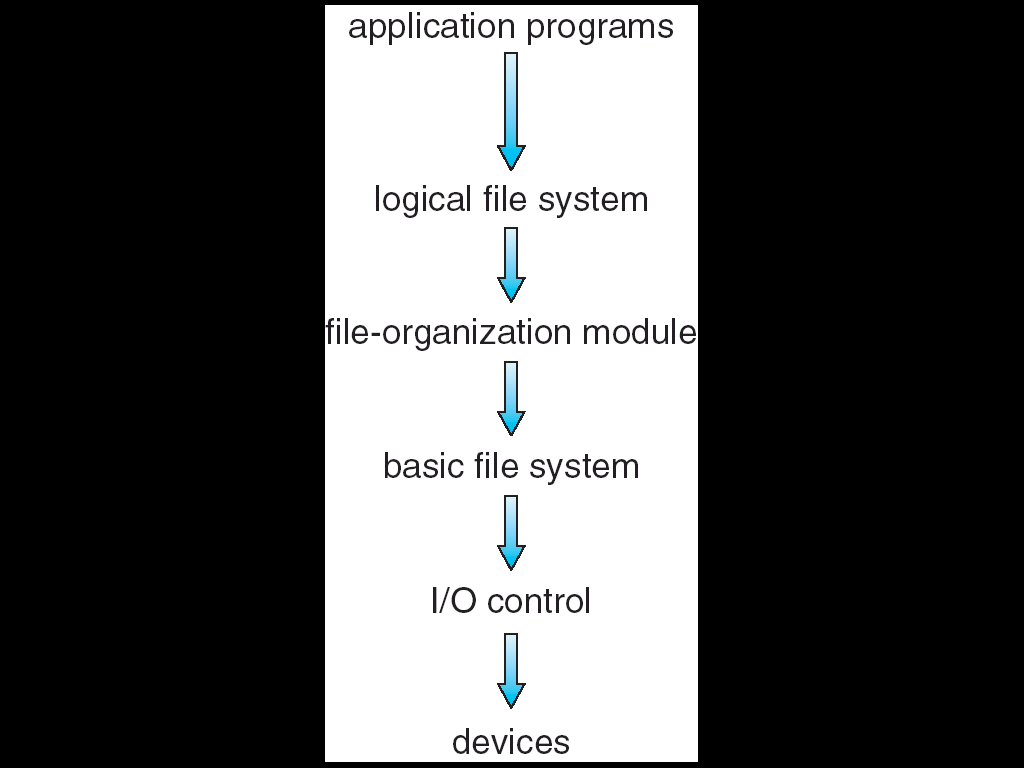
**a. How the file system should look to the user.**

**b. Selecting algorithms and data structures that must be created to map logical file**

system on to the physical secondary storage devices. The file system itself is composed of

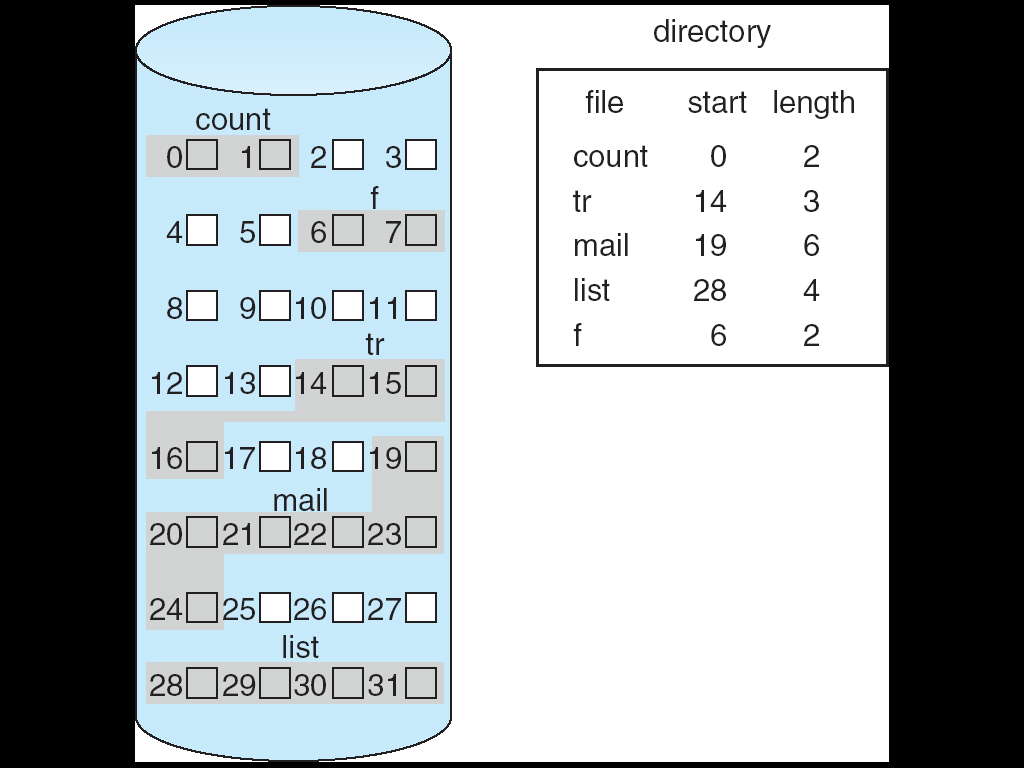
different levels. Each level uses the feature of the lower levels to create new features for use by higher levels.

**The following structures shows an example of layered design**



The lowest level is the i/o control consisting of device drivers and interrupt handless to transfer the Information between memory and the disk system. The device driver is like a translator. Its input is a high level command and the o/p consists of low level hardware specific instructions, which are used by the hardware controllers which interface I/O device to the rest of the system. The basic file system needs only to issue generic commands to the appropriate device drivers to read and write physical blocks on the disk. The file organization module knows about files and their logical blocks as well as physical blocks. By knowing the type of file allocation used and the location of the file, the file organization module can translate logical

1. **Explain contiguous allocation method with neat diagram**



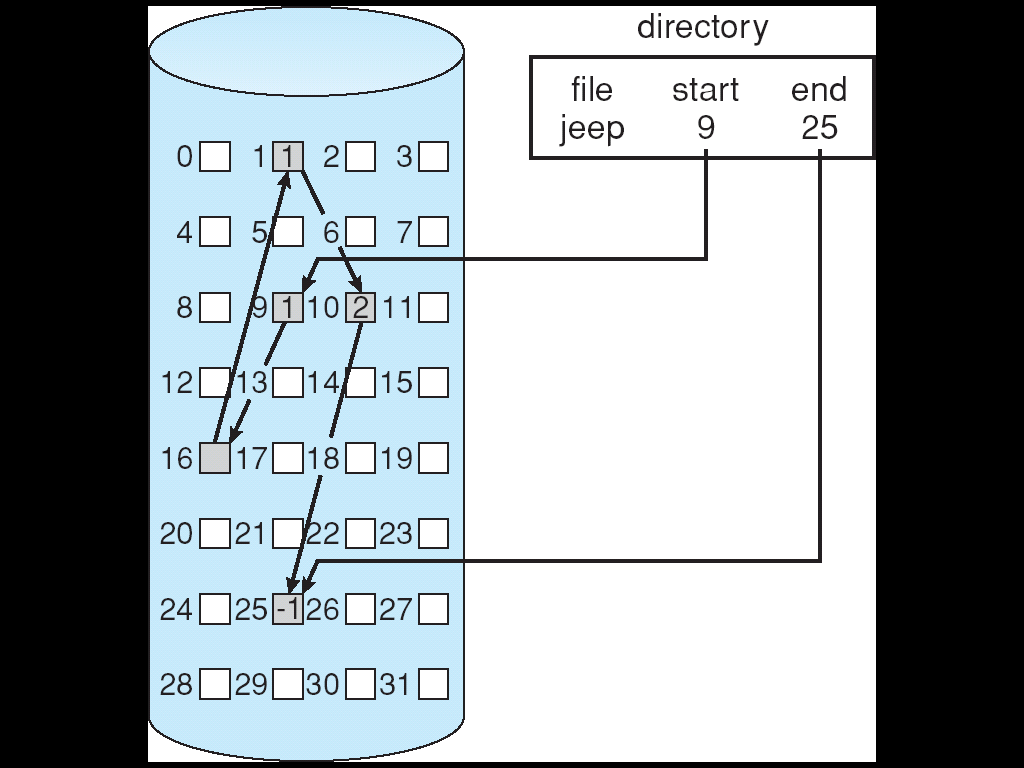
In this scheme, each file occupies a contiguous set of blocks on the disk. For example, if a file requires n blocks and is given a block b as the starting location, then the blocks assigned to the file will be: b, b+1, b+2,……b+n-1. This means that given the starting block address and the length of the file (in terms of blocks required), we can determine the blocks occupied by the file.  
The directory entry for a file with contiguous allocation contains

* Address of starting block
* Length of the allocated portion.

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. This 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.

1. **Explain Linked allocation method with neat diagram**



In this scheme, each file is a linked list of disk blocks which need not be contiguous. The disk blocks 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.

The file ‘jeep’ in following image shows how the blocks are randomly distributed. The last block (25) contains -1 indicating a null pointer and does not point to any other block.

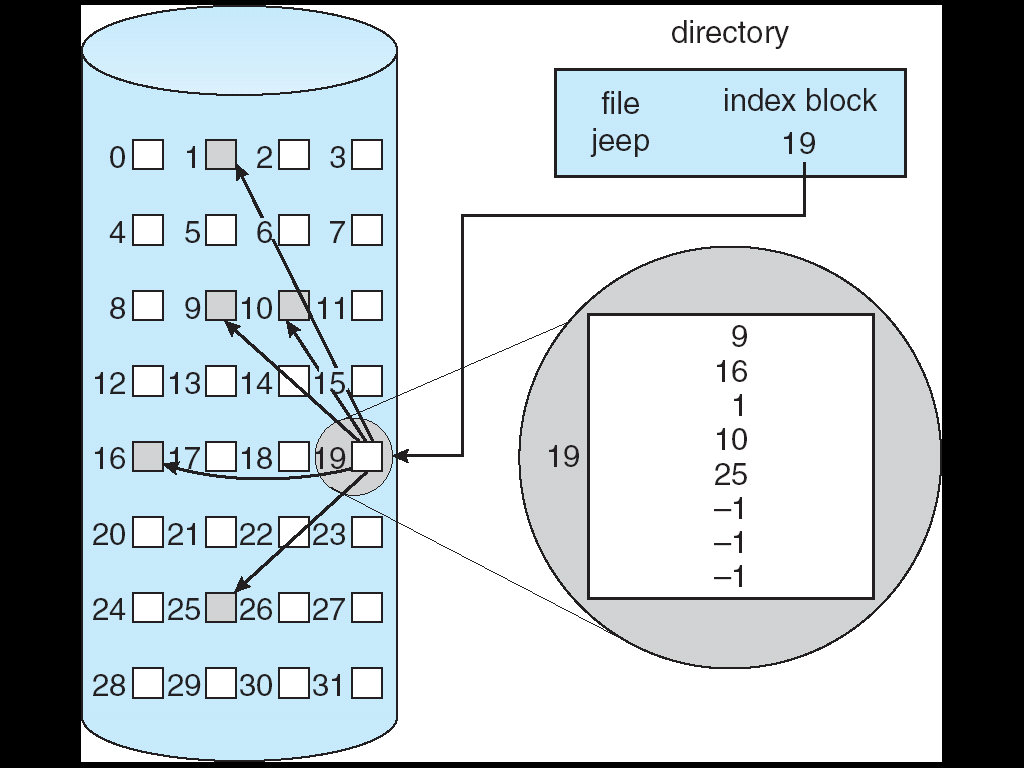
Advantages:

* 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. This makes it relatively better in terms of memory utilization.

Disadvantages:

* Because the file blocks are distributed randomly on the disk, a large number of seeks are needed to access every block individually. This makes linked allocation slower.
* It does not support random or direct access. We can not directly access the blocks of a file. A block k of a file can be accessed by traversing k blocks sequentially (sequential access ) from the starting block of the file via block pointers.
* Pointers required in the linked allocation incur some extra overhead.

1. **Explain Indexed allocation method with neat diagram**



In this scheme, a special block known as the Index block contains the pointers to all the blocks occupied by a file. Each file has its own index block. The ith entry in the index block contains the disk address of the ith file block. The directory entry contains the address of the index block as shown in the image:

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.

1. **Explain free space management in file system**

Since disk space is limited, we need to reuse the space from deleted files for new files, if possible. (Write-once optical disks only allow one write to any given sector, and thus such reuse is not physically possible.) To keep track of free disk space, the system maintains a free space list.

**Bit Vector**

Frequently, the free-space list is implemented as a bit map or bit vector .Each block is represented by 1 bit. If the block is free, the bit is 1; if the block is allocated, the bit is 0.

**Linked List**

Another approach to free-space management is to link together all the free disk blocks, keeping a pointer to the first free block in a special location on the disk and caching it in memory

**Grouping**

A modification of the free-list approach stores the addresses of n free blocks in the first free block. The first n-1 of these blocks are actually free. The last block contains the addresses of another n free blocks, and so on

