

Chapter 7

File Systems

--By R.G.B

File

- Resource for storing information
- Durable storage
- Properties of File
 1. Long term storage
 2. Sharable between processes
 3. Structure

File Descriptor

- **Abstract** indicator to file
- Points to file in the memory(primary)
- Non negative integer
- Index into a table maintained by the kernel that tracks which files are "opened" by a process for performing input/output.

File Naming

- A **filename** (also written as two words, **file name**) is a name used to uniquely identify a computer file stored in a file system.
- Different file systems impose different restrictions on filename lengths and the allowed characters within filenames

File Naming

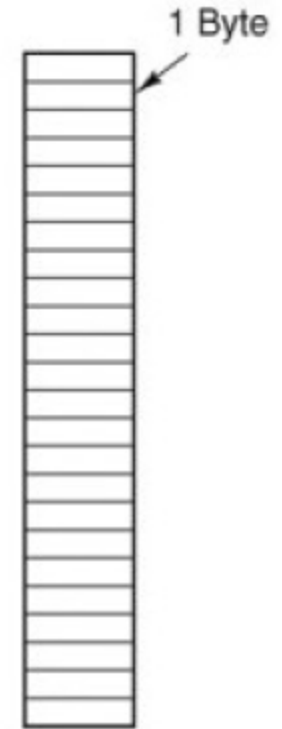
- A filename may include one or more of these components:
- **host** (or **node** or **server**) – network device that contains the file
- **device** (or **drive**) – hardware device or drive
- **directory** (or **path**) – directory tree (e.g., /usr/bin, \TEMP, [USR.LIB.SRC], etc.)
- **file** – base name of the file
- **type** (format or **extension**) – indicates the content type of the file (e.g., .txt, .exe, .COM, etc.)
- **version** – revision or generation number of the file

File Structure/Format

- File structure is a structure, which is according to a required format that operating system can understand.
- A file has a certain defined structure according to its type.
 1. Byte Sequence
 2. Record Sequence
 3. Tree

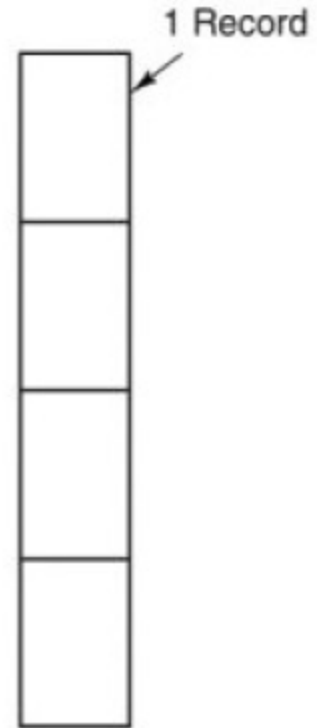
Byte Sequence

- Unstructured sequence of bytes
- OS does not know or care what is in the file
- Any meaning must be imposed by user-level programs
- Both UNIX and Windows 98 use this approach



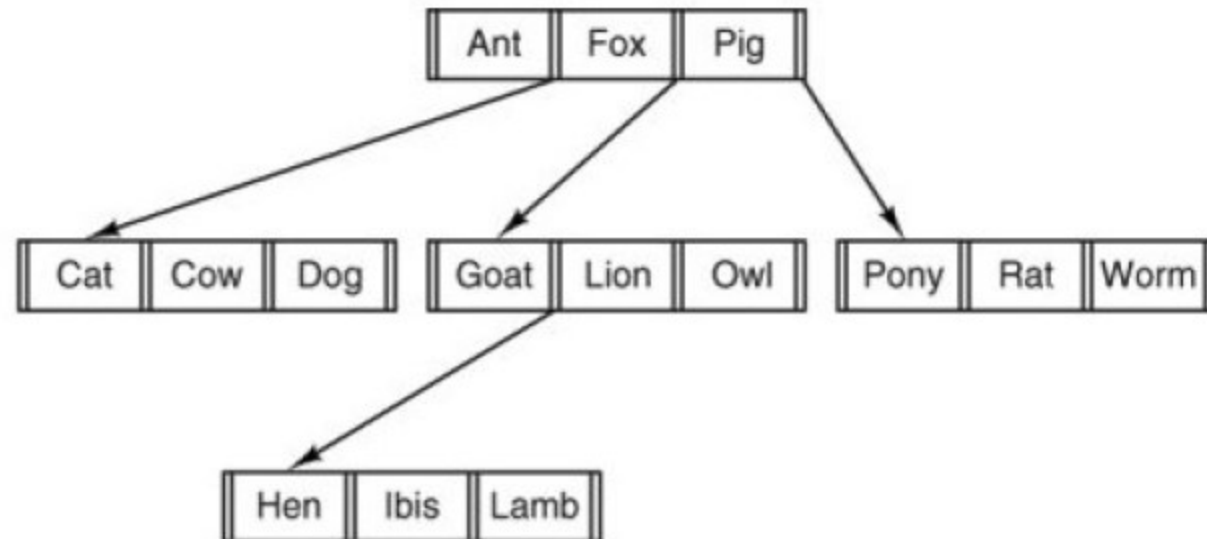
Record Sequence

- File is a sequence of fixed-length records, each with some internal structure
- Central to the idea of a file being a sequence of records is the idea that the read operation returns one record and the write operation overwrites or appends one record
- Usage : Main Frame Computer



Tree

- File consists of a tree of records ,Not necessarily all the same length
- Each containing a key field in a fixed position in the record
- The tree is sorted on the key field. to allow rapid searching for a particular key



File Types

- Ability of the operating system to distinguish different types of file such as text files source files and binary files etc...
- A **file format** is a standard way that information is encoded
- Specifies how bits are used to encode information
 1. Text file → Sequence of characters organized into lines
 2. Source file → Sequence of procedures and functions
 3. Object file → Sequence of bytes organized into blocks
- Many operating systems support many types of files

File Types

- Operating system like MS-DOS and UNIX have the following types of files

1. Ordinary files

- Files that contain user information
- May have text, databases or executable program
- Operations on such files like add, modify, delete or even remove the entire file

2. Directory files

- Files contain list of file names and other information related to these files

3. Special files

- Also known as device files
- Represent physical device like disks, terminals, printers, networks

File Types

- These files are of two types
- **Character special files** - data is handled character by character as in case of terminals or printers
- **Block special files** - data is handled in blocks as in the case of disks and tapes

File Attributes(Metadata)

- Metadata → data that describes other data(in this case file)
- Meta data /Attributes is provided by os
- Attributes varies with O.S
- Some of the attributes are Listed below
 1. Protection→ who can access file?
 2. Password → Password for accessing
 3. Creator → id of user who created the file
 4. Owner → current owner/s

File Attributes(Metadata)

- 5. Read only flag → 1 read/write and 0 readonly
- 6. Hidden flag → 0 unhide and 1 hidden
- 7. System flag → 0 normal and 1 system
- 8. Archive flag → 0 backed up and 1 not backed up
- 9. Ascii/binary flag → 0 Ascii file and 1 binary file
- 10. Random access flag → 0 sequential and 1 random
- 11. Temporary flag → 0 normal and 1 temporary
- 12. Lock flag → 0 unlocked and 1 locked

File Attributes(Metadata)

- 13. Record length → no bytes in record
- 14. Key position → offset of key within each record
- 15. Key length → no of bytes in key field
- 16. Creation time
- 17. Time of last access
- 18. Time of last change
- 19. Current size
- 20. Max size

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
12. Lock

File Access Methods

- File access mechanism refers to the manner in which the records of a file may be accessed
- There are several ways to access files
 1. Sequential access
 2. Direct/Random access
 3. Indexed sequential access

Sequential Access

- Simplest Access mechanism
- Processed in sequential way
- One record process after the other
- This access method is the most primitive one
- Example: Compilers usually access files in this fashion

Random Access

- Random access file organization provides, accessing the records directly.
- Each record has its own address on the file with by the help of which it can be directly accessed for reading or writing.
- The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.
- Suited for DB

Index Sequential Access

- This mechanism is built up on base of sequential access.
- An index is created for each file which contains pointers to various blocks
- Index is searched sequentially and its pointer is used to access the file directly

Sharing

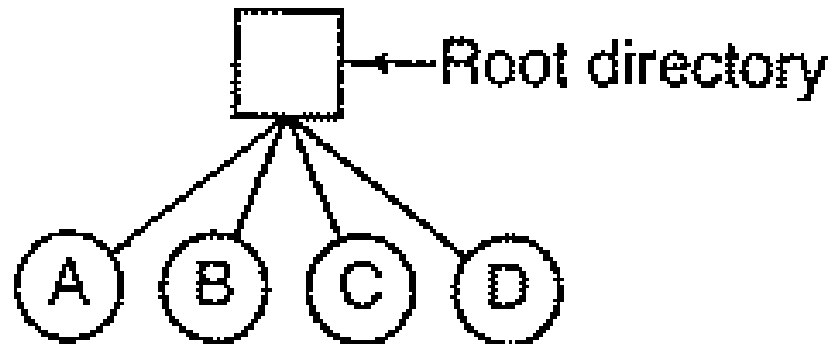
- **File sharing** is the practice of distributing or providing access to digital media To / from remote system
- File sharing may be achieved in a number of ways
- Peer to peer sharing
- File sync and sharing services

Directories

- Files that keep track of files
- Two types
 1. Single level Directory system

Single level Directory system

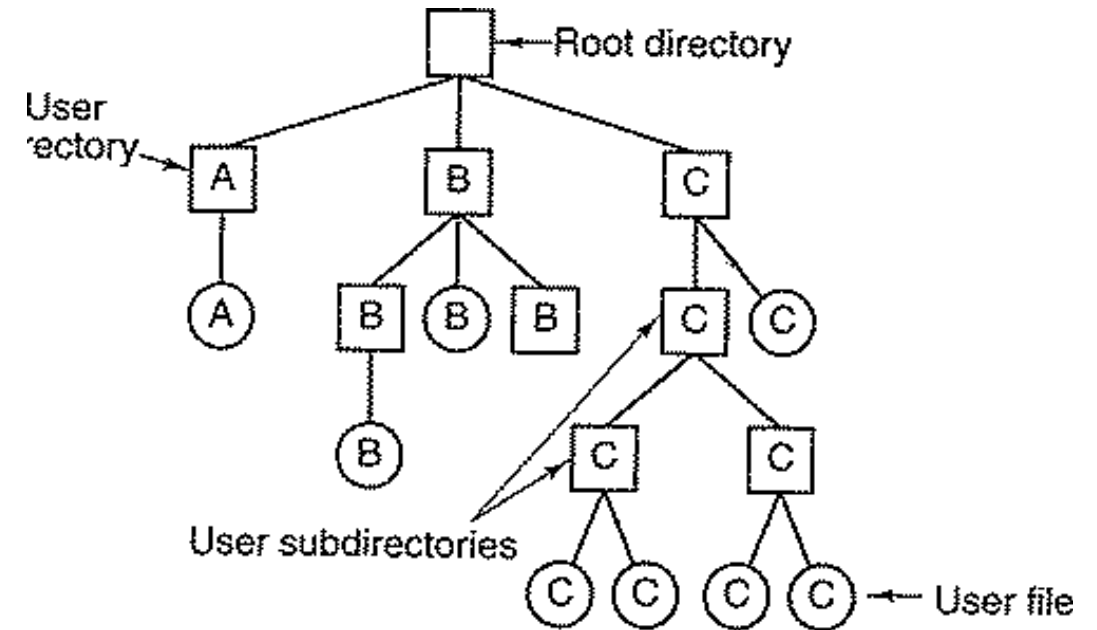
- Simplest form of directory system is having one directory containing all the files
- Sometimes it is called the **root directory**
- **Example** : CDC 6600



Hierarchical Directory System

More than one level of directories

File and/or folders inside directories



Directory Operations

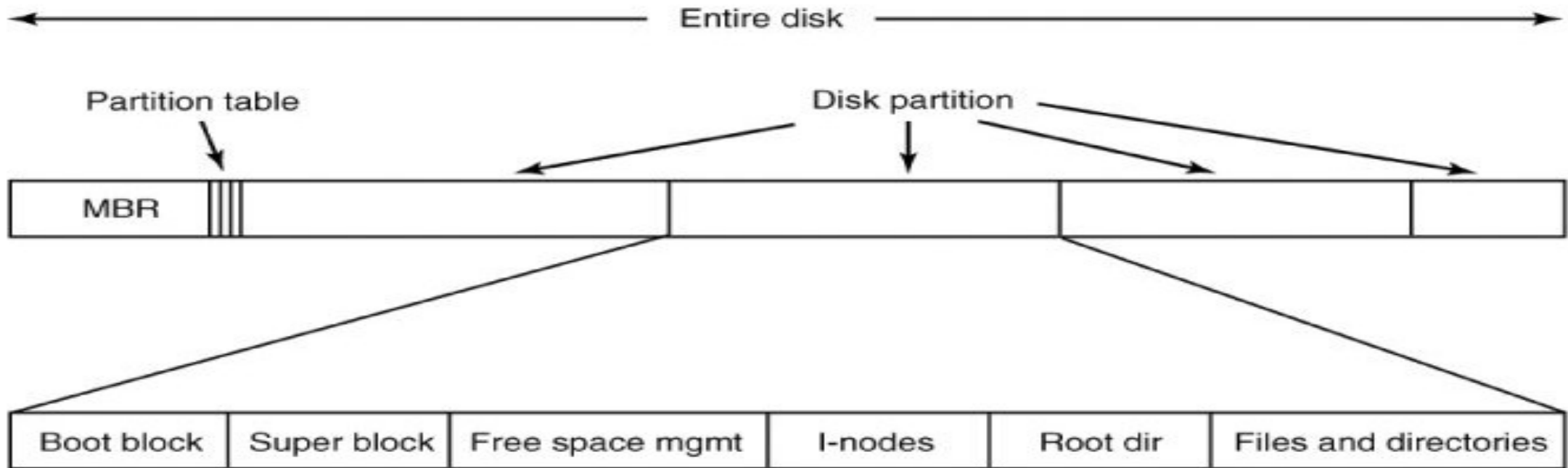
- Create
- Delete
- Opendir
- Closedir
- Readdir
- Rename
- Link
- Unlink

File System Implementation

System Layout

- File systems are stored on disks.
- Most disks can be divided up into one or more partitions, with independent file systems on each partition.
- Sector 0 of the disk is called the MBR (Master Boot Record) and is used to boot the computer.
- The end of the MBR contains the partition table.
- This table gives the starting and ending addresses of each partition.
- One of the partitions in the table is marked as active.

System Layout



File Allocation Methods/ File Implementations

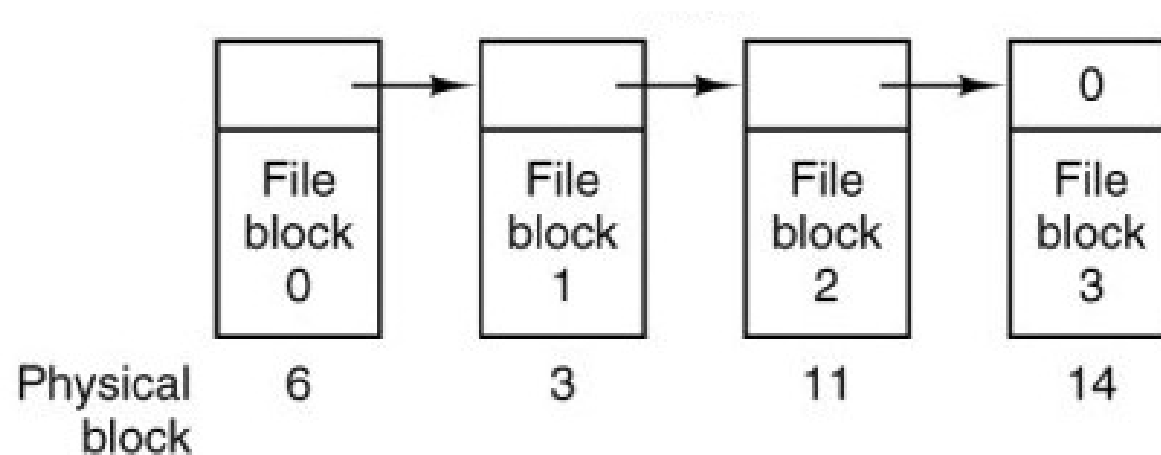
1. **Contiguous Allocation**
2. **Linked List Allocation**
3. **Linked List Allocation Using a Table in Memory**
4. **I-Nodes**

Contiguous Allocation

- Occupy a set of contiguous addresses on a disk
- It store each file as a contiguous run of disk blocks
- Disk with 1-KB blocks, a 5-KB file would be allocated 5 consecutive blocks
- Sequential and direct access is supported
- Simple to implement
- High Read Performance
- Ex: CD/ DVD - ROM

Linked List Allocation

- keep each file as a linked list of disk blocks
- The first word of each block is used as a pointer to the next one.
- The rest of the block is for data



Linked List Allocation

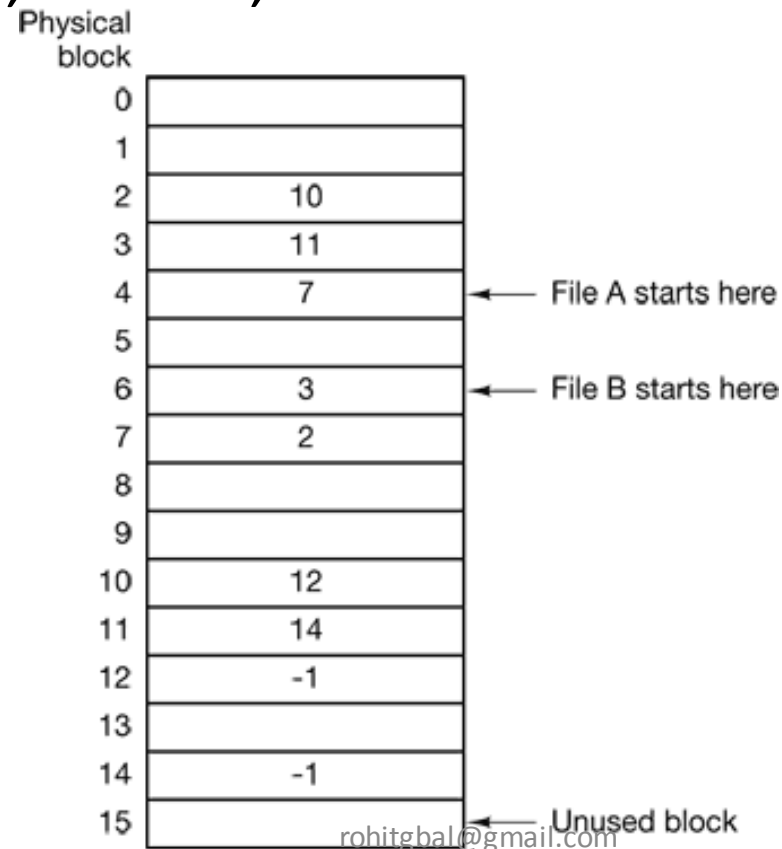
- No space is lost to disk fragmentation
- The major problem with linked allocation is that it can be used only for sequential access files
- Find ith block of a file, we must start at the beginning of that file, and follow the pointers until we get the ith block
- Linked list allocation is reliability(Pointer Lost)

Linked List Allocation Using a Table in Memory

- Disadvantages of the linked list allocation can be eliminated by taking the pointer word from each disk block and putting it in a table in memory
- Table in main memory is called a **FAT (File Allocation Table)**

Linked List Allocation Using a Table in Memory

- File *A* uses disk blocks 4, 7, 2, 10, and 12, in that order, and file *B* uses disk blocks 6, 3, 11, and 14, in that order

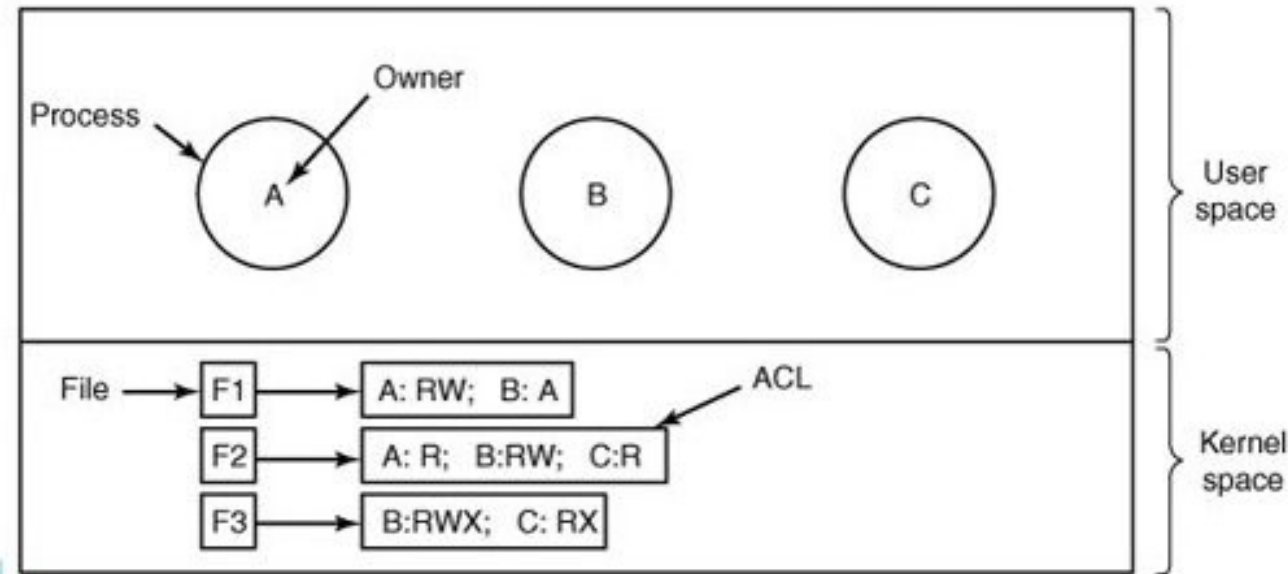


I Nodes

- All location all pointers are brought together into one location called **Index block**
- Each file has its own index block, which is an array of disk-block addresses
- The i th entry in the index block points to the i th block of the file
- The directory contains the address of the index block

Access Control List(ACL)

- ACL consists of set of entries
- Specifies Access permissions to files for users
- Permissions → Read(R), Write(W), Execute(X)



Access Control List(ACL)

- Access control is list of permissions
- File access is possible according to permissions
- Each file has its access permission
- File permission is stored in its metadata
- Directories ACL will effect all files inside it