

Lab Report No: 04

Lab Report Name: File operation and permission.

Name : Golam Kibria Tuhin

ID : IT-18015

Objective:

- To explain the function of file systems.
- To describe the interfaces to file systems.
- To discuss file-system design tradeoffs, including access methods, file sharing, file locking and directory structures.
- To explore file-system protection.
- To describe the details of implementing local file systems and directory structures.
- To describe the implementation of remote file systems.
- To discuss block allocation and free-block algorithms and trade-offs.

Question-01: What is File?

Answer: File is a set of collections of the related information that are recorded on the secondary memory.

Question-02: What are File Operations and File Permission in the Linux Operating System?

Answer: Operating system provides system called to write, delete, truncate, reposition, read and create files. There are about six basic file operation within in the operating system.

Creating a file: Two steps necessary to creating a file. First, space in the file system. Second, an entry for the new file must be made in the directory.

1. **Writing a file:** For writing to a file, we make a system call specify about both the name of the file and the information to be written to the file.
2. **Reading a file:** For reading from a file, we use a system call read.
3. **Repositioning inside a file:** Repositioning file operation is termed as 'file seek.'
4. **Deleting a file:** For deleting any file, we can use delete operation.
5. **Truncating a file:** We can erase all information contents of a file but without the name of attributes.

File Permission:

Each file and directory has three user based permission groups:

- I. **Owner:** The Owner permissions apply only the owner of the file or directory, they will not impact the actions of other users.
- II. **Group:** The Group permissions apply only to the group that has been assigned to the file or directory, they will not effect the actions of other users.
- III. **All user:** The All Users permissions apply to all other users on the system, this is the permission group that you want to watch the most.

Every files and also directory in our UNIX or Linux system have following 3 permissions that are defined below

1. Read
2. Write
3. Execute permission

Read (r): Read operation gives us just the authority for opening and reading a file.

Write (w): Write operation gives us just the authority for modify of the contents any file.

Execute (x): Execute operation gives us just the authority to run a file.

Question-03: Implementation of a File Operation and a File Permission and also File Permission in the Linux Operating System?

Answer: Numerous on-disk and in-memory configurations and structures are being used for implementing a file system. These structures differ based on the operating system and the file system but applying some general principles. Here they are portrayed below:

- IV. A boot control block usually contains the information required by the system for booting an operating system from that volume. When the disks do not contain any operating system, this block can be treated as empty. This is typically the first chunk of a volume. In UFS, this is termed as the boot block; in NTFS, it is the partition boot sector.
- V. A volume control block holds volume or the partition details, such as the number of blocks in the partition, size of the blocks or chunks, free-block count along with freeblock pointers. In UFS, it is termed as superblock; in NTFS, it is stored in the master file table.
- VI. A directory structure per file system is required for organizing the files. In UFS, it held the file names and associated 'inode' numbers. In NTFS, it gets stored in the master file table.
- VII. The FCB contains many details regarding any file which includes file permissions, ownership; the size of file and location of data blocks. In UFS, it is called the inode. In NTFS, this information gets stored within the master file table that uses a relational database (RDBM) structure, using a row per file.

Conclusion:

The file system resides permanently on secondary storage, which is designed to hold a large amount of data permanently. The most common secondary-storage medium is the disk. Any file-system type can have different structures and algorithms. A VFS layer allows the upper layers to deal with each file-system type uniformly. Even remote file systems can be integrated into the system's directory structure and acted on by standard system calls via the VFS interface.