Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 04

Course code: ICT- ICT-3110

Course title: Operating Systems Lab

Date of Performance:15-09-20

Date of Submission:27-09-20

Submitted by

Name: Nusrat Jahan Jui

ID:IT-18039

3rd year 1stsemester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment no : 04

Experiment Name : File operation and permission.

Theory:

A *file* is a named collection of related data that appears to the user as a single, contiguous block of information and that is retained in *storage*. Just likely in Linux operating system, everything is organized in the form of files and directories. Linux is a clone of UNIX, the multi-user operating system which can be accessed by many users simultaneously. Linux can also be used in mainframes and servers without any modifications. But this raises security concerns as an unsolicited or malign user can corrupt, change or remove crucial data. For effective security, Linux divides authorization into 2 levels: Ownership and Permission.

Every file and directory on your Unix/Linux system is assigned 3 types of owner: user , group , other.

When the big question arises how does Linux distinguish between these three user types so that a user 'A' cannot affect a file which contains some other user 'B's' vital information/data. It is like you do not want your colleague, who works on your Linux computer, to view your images. This is where Permissions set in, and they define user behavior**.**

Implementation:

File operation:

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:

- 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.

- 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 free-block pointers. In UFS, it is termed as superblock; in NTFS, it is stored in the master file table.

- 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.

- 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.

Some basic file operations are given below:

1.Copying files: (cp *file1 file2)* is the command which makes a copy of file1 in the current working directory and calls it file2**.**

2.Moving files: (mv *file1 file2)* moves or renames file1 to file2.

3.Removing files and dictionaries: To delete or remove a file, use the rm command and rmdir command to remove a directory.

4.Searching the contents of a file: Using less, you can search though a text file for a keyword.

5.Clear screen: (clear) Before you start the next section, you may like to clear the terminal window of the previous commands so the output of the following commands can be clearly understood.

Permission:

Each file and directory has three user based permission groups:

Owner: The Owner permissions apply only the owner of the file or directory, they will not impact the actions of other users.

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.

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.

Each file or directory has three basic permission types:

- **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.

- **Write:** The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.

- **Execute:** In Windows, an executable program usually has an extension ".exe" and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code (provided read & write permissions are set), but not run it.

The easiest way to view the permissions of files in a given directory is to run:

ls –l <path to directory>

If you want to view the permissions in your current directory, leave out the directory name at the end:

ls –l

Discussion :

File permission can lead us to work unitedly and smartly. Linux is a multi-user operating system. Thus, we need the concepts of file ownership and permissions to ensure the security of files, as well as the OS. When working with permissions of system files, you should be very careful as wrong permissions can prevent programs from working correctly.