**Faculty of Computing**



**Operating Systems**

**LAB # 04**

**Instructor**

Ayesha Akram

Faculty of Computing,

Riphah International University,

Islamabad

**File Editors**

1. **Create file using pico editor**

Type following command **pico notes.** With this command an editor will be opened. Enter text into the file. Press **Ctrl-O** to **save** the file, and **Ctrl-X** to **exit** from the editor.

1. **Create file using nano editor**

Type following command **nano notes** and write any text in it**.** To open a file for editing, type nano filename.txt If the file doesn’t exist, Nano will create it. After editing, press Ctrl + O (for "write out"), then press **Enter** to save the file. Press Ctrl + X to exit. If there are unsaved changes, it will prompt you to save.

**Commands**

1. **cat command**

In order to view an existing file use **“cat”** command. Now press following command on shell **cat notes** (file name is notes)the contents of file notes will be visible. It is used to **view**, **concatenate**, and **create files** in the terminal. It is short for **concatenate**, as it was originally designed to concatenate and display the contents of files.

**Syntax:**

cat filename ( view file)

cat filename filename1 (to concatenate files)

cat >filename3 (to create a file)

ctrl+d ( to save file or end of terminal)

# cp command

|  |  |
| --- | --- |
| **Command** | **Description** |
| **cp** | Creates a new copy of existing file. |
| **Example** | **cp oldFile newFile** |

cp -i (ask first for over write)

To copy file from the current directory to another directory the general format is

**cp oldfile newfile directoryname .** It will copy both files into another directory.

# mv command

|  |  |
| --- | --- |
| **Command** | **Description** |
| **mv** | Renames a file or moves it from one directory to another. |
| **Example** | **mv oldFile newFile** |

**mv** command can also be used to renamea file. The **oldFile** specifies the existing file you want to rename. **newFile** specifies the new name to use for the file. **mv fileName directory/newName** The **fileName** specifies the file you want to move, **directory** specifies the directory into which you want to move the file. **newName** specifies the new name to use for the file.

# Complete Information about a File/Directory

When I entered following command on my command prompt I got following results:

ls –l notes

## 

The following information is printed in the given order:

1. File permissions
2. File links
3. Owner of file
4. Group
5. Size of file
6. Last change
7. Time of change
8. File/directory name

**Note:**

You must remember the sequence in which the above information displayed.

# File and Directory Security

Files/Directories can be created by setting permissions, allow people to **read, write, or execute your file**. Each file on the machine divides the users of machine into three categories:

* The file’s **owner** (who creates the file)
* A **group** of users
* **Other** users

Actually, there is one more type of user; that’s **super user**. The system administrator may be the only **super user**, but often several people have access to the **super user** password. Anyone logged in as the **super user** has total access to every file directory in the system.

# Types of Access

There are three types of access:

## Read

* Write
* Execute

# File Permissions

If a file has **read** permission, it can be examined at a terminal, printed, viewed by an editor, and so on.

If it has **write** permission, the contents of the file can be changed (for example, by an editor), the file can be overwritten or deleted.

If it has **executed** permission, and is binary program or a shell script, that program can be run (but copied only if it also has **read** permission).

Having a type of access is referred to as having **access permission**. You can change the access permission for your own file. For example, if you do not want anyone else to access a file, you can remove read access for anyone but you. If you want other users in your group to be able to write to group of files, you can extend write permission to group

Each of the three user types (the owner, the group, and others) can have any combination of the three access types for each file or directory.

# Directory Permissions

Directories have permissions modes that work in a similar manner to file permission modes. However, the directory access permissions have different meanings:

**Read:** The read (**r**) permission in a directory means you can use the **ls** command to the filenames.

**Write:** The write (**w**) permission in a directory means you can add remove files from that directory.

**Execute:** The execute (**x**) permission in a directory means you can use the **cd** command to change to that directory.

**Note:** Your installation has default set up for all newly created files and directories. You can check your default access through the **ls -l** command.

# Access Specification

When you create a file or directory it come into existence with some default access specifications. It may give all access permissions to the owner, just read and write permissions to the group, and just read permission to everyone, or there may be any situation.

Following figure shows how different groups and characters represent access permissions.

**-**rw-r- - r- - (file)

drwx rw-r-- (directory)



user

group

others

The first character indicates whether it is a **file** or **directory**.The next nine places are divided into three sets, each of length 3. The first set of three indicates the **owner access**, the next set of three indicates the **group access**, and the final set of three indicates the access for everybody else. The maximum access is represented by ***rwx***, indicating **read**, **writes**, and **execute**. Whenever a dash (─) appears, an access permission has not been given.

# Checking Access

The access privileges in a given directories can be checked by giving command **ls -l**.

**Example**

**-**rw-r- - r- - 1 saleem stud 700 June 19 08:0 data.file Access links owner group size last change name

In this example, the first character is −, indicating that it is a file. The owner’s group indicates that the owner i.e. Salem can read and write, but could not execute. Any group member can read the file but do not execute. Also some other person can read this file but do not write it or can execute it.

# Groups

It is the authority of administrator to create a group of users. Each group has number of users, and given a unique name. Each group has also a unique number **(group id),** and each user has also a unique number, called **user-id or UID.**

# id command

|  |  |
| --- | --- |
| **Command** | **Description** |
| **id** | It gives user’s name together with the groups they are member, both names and numbers, and the users.  user-id and current group-id. |

Example

## id (enter)

uid=275(john1), group=50(staff)

id chris

*uid=145(charis) gid=12(ugrads) groups=12(ugrads), 417(proj)*

This shows that **chris** is a member of groups **ugrads**, and **proj**, with **GID** numbers 12 and 417 respectively. Currently chris is allotted to group ugrads.

The access privileges can be changed. A command **chmod** is used to change the access privileges.

# chmod command:

|  |  |
| --- | --- |
| **Command** | **Description** |
| **chmod(change mode)** | Change access permission for one or more files.  **chmod user-type [operations][permissions] file-list**  i.e chmod o+w notes |

|  |  |
| --- | --- |
|  | **User-type**  **u** User or owner of file.  **g** Group that owns the file.  **o** Other.  **a** All three user types. |
|  | **Operations**  ***+*** Add the permission.  **-** Remove the permission.  **=** Set exact permission, all other permission. Reset. |
|  | **Permissions**  **r** Read permission.  **w** Write permission.  **x** Execute(run) permission. |

However, you can specify the new mode as a three-digit number that is computed by adding together the numeric equivalents of the desired permission. Following table shows numeric value assigned to each permission letter.

|  |  |  |
| --- | --- | --- |
| **Owner** | **Group** | **Other** |
| **r w x** | **r w x** | **r w x** |
| 4 2 1 | 4 2 1 | 4 2 1 |

**Example**

Change access permission to allow the read, write, and execute permission to all users.

**chmod 777** file/directory name

**Class Tasks**

1. Rename file atomic-notes with at-notes in the books directory.
2. Copy this file to sub-directory nuclear under the directory physics.
3. Removed Write(w) permission for owner (user) from file result.comp.

Group and other get execute access for file stock.

Set group access for reading(r) and execute(x) of myfile.

1. Create a directory called l**ab4** and create three files say quiz, report and viva inside the directory

## Now try to set the following rights;

## -rw- r-- r- quiz

-**rw- -w- r- - report**

## -rwx rwx r-x viva

Also try to change access permission of **lab4** directory as well and see how it affects.