

Unit 11 : Linux user and file management

Lesson 1 : User management in Linux

1.1. Learning Objectives

On completion of this lesson you will be able to know:

- ❖ About different users in Linux
- ❖ How to Log in
- ❖ How to create user
- ❖ How to delete user
- ❖ How to change password of user
- ❖ How to Log Out

1.2. Types of users in Linux

Supper user or root user

Super or root user is a special kind of user account that holds all kinds of permissions to do any alteration to a programs or services of Linux. Especially this kind of user account is used for system administration, who can control and limit the access of other User's

Types of users in Linux

System User

System User is created by default by the OS. This type of user is similar to normal user but gets more privileges and access to secure programs which normal user doesn't have.

Normal User

These are the users which have been created by the Root and have limited access to the resources and need permission from Root to access any secure resources and services.

1.3. Logging In

Logging In

Logging in is the process by which individual [access](#) to a [computer system](#) is controlled by [identifying](#) and [authenticating](#) the [user](#) through the [credentials](#) presented by the user. After you connect to the Linux system, a message similar to the one shown below appears at the terminal:

```
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686
```

Login:

Each user has user name, which has to be entered when, the login: prompt appears. After you enter your login name, you are asked to enter your password. If the login

name entered does not match any of the user names in the stored file, the login message is displayed again. When a valid user name is entered at the terminal, the **[user_name@localhost current_directory_name]** \$ symbol is displayed on the screen. This is the shell prompt, in which user_name is the user's login name and current_directory_name is the user's current working directory.

The administrator assigns each user a HOME directory when a new login account is created. When you log in the terminal, you are taken directly into your HOME directory. In Linux operating system, login names (user names) are usually the names of users, and their HOME directory usually, although not necessarily, has the same name. For instance, if your user name is Odroho and your HOME directory name is also Odroho, when you logging in, you will see the following prompt on the screen:

```
[Odroho@localhost Odroho]$
```

You can now start working on Linux.

User name and password are case-sensitive. The total login process appears like the one shown below:

```
Red Hat Linux release 9 (Shrike)
```

```
Kernel 2.4.20-8 on an i686
```

```
Login: Odroho
```

```
Password: [user enters password here]
```

```
Last login: Fri Dec 13 12:18:02
```

```
[Odroho@localhost Odroho] $ _
```

1.4. Creating user account with password in Linux

Most system administration task requires that you login as root. To create a user account you use the **adduser** command

Creating user account with password.

```
[root@localhost root]# adduser Odroho
```

```
[root@localhost root]# passwd Odroho
```

```
New password:
```

```
BAD PASSWORD: it is based on a dictionary word
```

```
Retype new password:
```

```
passwd: all authentication tokens update successfully
```

1.5. Deleting user account

You need to use the **userdel** command to delete a user account and related files from user account. The **userdel** command must be run as root user. The syntax is as follows:

Example

```
[root@localhost root]# userdel Odroho
```

```
[root@localhost root]# userdel -r Odroho
```

The second command will remove all files along with the home directory itself and the user's mail spool. Please note that files located in other file systems will have to be searched for and deleted manually.

1.6. Security for the Linux user

Linux allows an additional measure of security by allowing you to possess a password associated with your login name. To log in using a password-protected user name, you should enter not only the user name but also the password. Passwords are not displayed on the screen while they are being entered.

Changing the user password

A user can change his password with the **passwd** command. The steps followed by the user, Odroho, to change his password are depicted below.

Example

```
[Odroho@localhost Odroho]$ passwd
Changing password for Odroho
(current) UNIX password: [user enters old password here]
New UNIX password: [user enters new password here]
Retype new UNIX password [user re-enters new password here]
passwd: all authentication tokens updated successfully
[Odroho@localhost Odroho]$
```

The **passwd** command asks for the old password to ensure that only the authorized user is trying to change the password. If Linux does not recognize the old password, it displays a "passwd: Authentication failure" message, and the Linux prompt appears on the screen.

Example

```
[Odroho@localhost Odroho]$ passwd
Changing password for Odroho
(current) UNIX password: [user enters the old password]
passwd: Authentication failure
[Odroho@localhost Odroho]$
```

Example

```
[Odroho@localhost Odroho]$ passwd
Changing password for Odroho
(current) UNIX password: [user enters the old password]
New UNIX password: [user enter a new password]
Retype new UNIX password: [user enters the wrong password]
Sorry, passwords do not match
New UNIX password:
```

When you change a password, the new password should differ from the old password by at least three positions. In Linux, a password should be at least six characters long it cannot be the same as the user's login name. The new password cannot be the same as the old password.

The **root** user has the authority to change the password of any user of the Linux system. On executing the **passwd** command followed by a user name, the **root**

Operating System

user is allowed to change the password for that user.

```
[root@localhost /root]# passwd Odroho
Changing password for Odroho
New UNIX password:                [the root user enters a new password]
Retype new UNIX password:         [the root user re-enters the new password]
passwd: all authentication tokens updated successfully
[root@localhost /root]#
```

The **root** user is the administrator of the Linux operating system. The prompt of the **root** user is denoted by a # sign and not the \$ sign, as in the case of normal users.

1.7. Logging Out

Once you logged on to the system, your work session continues until your instruction to the shell to terminate the session. Typing **exit** or **logout** at the command prompt will end your current Linux session. The system then displays the **login:** prompt on the screen.

In order to maintain the security of files, it is recommended to log out from the terminal.

Linux user and file management

1.8. Exercises

1.8.1. Multiple choice questions

- a. Which is the command used to create a user?
- (i) passwd
 - (ii) userdel
 - (iii) adduser
 - (iv) addusr
- b. How many types of users are in Linux?
- (i) Three
 - (ii) Four
 - (iii) Five
 - (iv) Six
- c. Which user can change the password of any user?
- (i) root
 - (ii) file owner user
 - (iii) group owner user
 - (iv) Odroho

1.5.2. Questions for short answers

- a. How do you logging in into a Linux terminal?
- b. How do you logging out from a Linux terminal?
- c. Create a user after your name using password.
- d. Change the password of the user that you have created.

1.5.3. Analytical questions

- a. Classify users. Discuss each of them.
- b. Discuss about Logging In and Logging out procedures.

Lesson 2 : File System and Directory Structure

2.1. Learning Objectives

On completion of this lesson you will be able to know:

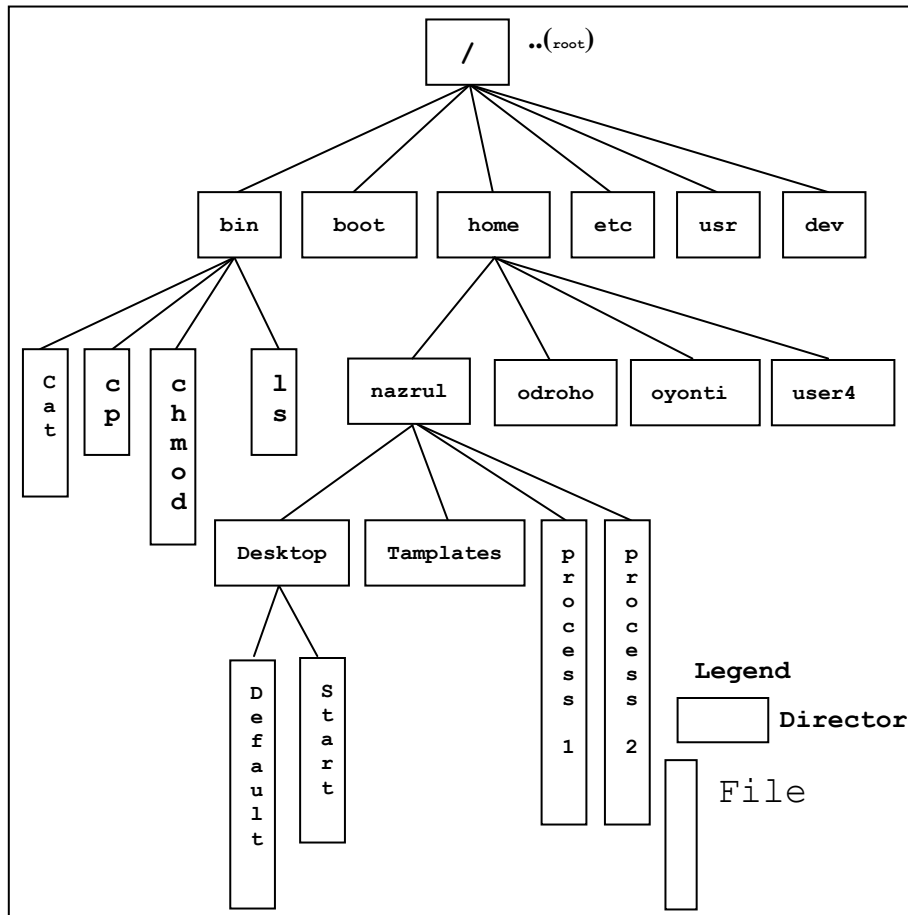
- ❖ File hierarchy of Linux operating system
- ❖ About the file naming convention of Linux
- ❖ Convenient way of specifying the path name
- ❖ About file Types

2.2. Linux File Systems

A file system is used to control how information is stored and retrieved.

The UNIX file system has a hierarchical structure and files can be stored under directories. The directories on the disk can be created to store files containing data of similar nature. The user decides the directory name and the files in a directory. Linux follows the UNIX file system convention.

A Sample Linux File Hierarchy



A file system is used to control how information is stored and retrieved.

Linux user and file management

All the files are stored on the disk under one main directory called the **root** directory. The **root** directory has been further sub-divided into directories-bin, boot, home, usr, etc, and dev. In each directory, files containing related data can be stored. The administrator of the Linux system may place all the HOME directories of the users under the */home* directory. For example, the nazrul directory is the HOME directory for the user, nazrul. He will store all his files in his HOME directory, nazrul, or create new directories under it.

The directory nazrul contains two files, namely process1 and process2, and two directories, Desktop and Templates. The Desktop directory contains two files, namely, Default and start.

In the file system hierarchy, a file is referred to by its path name, which is made up of the file name, preceded by the name of the directory containing the file. The path name may have a set of directories, one directory containing another, until the '/' directory is reached. The file name and the different directory names contained in the path are separated by the '/' symbol.

For instance, in the previous figure, the full path of the file *start* can be given as: */home/nazrul/Desktop/start*. Similarly, the path for the file *ls* is */bin/ls*. As you notice, the '/' is a special character, therefore, it cannot be included in a file or directory name.

The following are some of the directories under the / directory. Each of these directories has been organized to store a specific type of file.

/bin : All the executable binary programs (file) required during booting, repairing, files required to run into single-user-mode, and other important, basic commands viz., cat, du,df, tar, rpm, wc, history, etc.

/dev : Contains device files for all the hardware devices on the machine e.g., cdrom, cpu, etc

/etc : Contains Application's configuration files, startup, shutdown, start, stop script for every individual program.

/lib : The Lib directory contains kernel modules and shared library images required to boot the system and run commands in root file system.

/home : Home directory of the users. Every time a new user is created, a directory in the name of user is created within home directory which contains other directories like Desktop, Downloads, Documents, etc.

/usr : Contains executable binaries, documentation, source code, libraries for second level program.

/var : Stands for variable. The contents of this file are expected to grow. This directory contains log, lock, spool, mail and temp files.

2.3. File Naming Convention in Linux

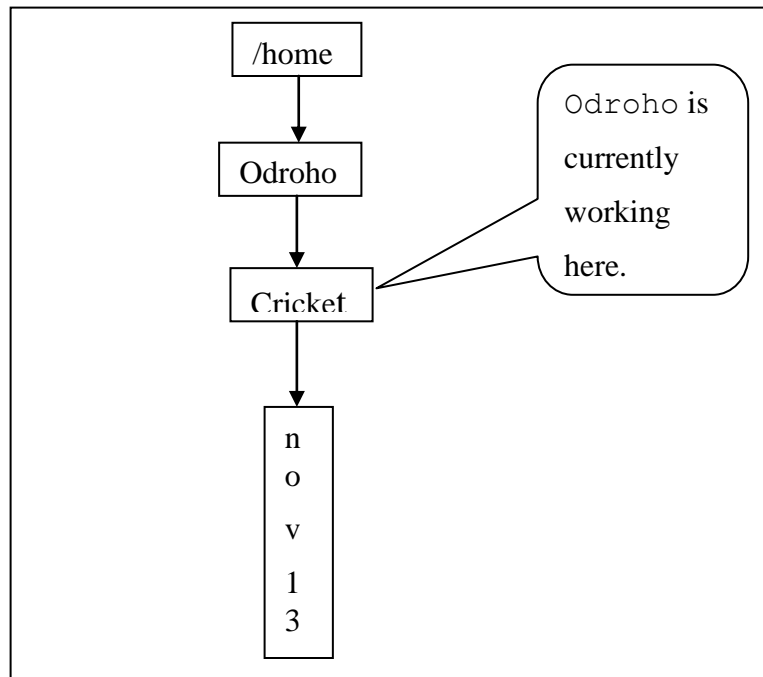
In Linux, file names:

- Can be up to 256 characters long
- Can contain special characters, except '/'
- Can contain both upper-case and lower-case alphabets
- Are case-sensitive
- Should not have a blank or a tab

2.4. Relative Path Names

Relative Path Names

The current working directory is the directory in which you are currently located. You can refer to it as your current directory or your current working directory. While referring to a file, instead of specifying the full path of the file, you can specify the path in relation to your current directory. Consider the following sample directory structure.



A Sample Directory Structure.

Figure: A Sample Directory structure.

For instance, let's say that the user, Odroho, is working in the directory, /home/Odroho. Odroho wants to access the file, nov13, under the /home/Odroho/Cricket directory. If he accesses this file using the full path name, Odroho would specify the path as /home/Odroho/Cricket/nov13. Alternatively, Odroho can specify the relative path, Cricket/nov13. This is a much more convenient way of specifying the path name.

Thus, notice that when you start the path name with a '/', Linux takes the path name as an absolute path name or the full path name. Whenever the path name starts with a character, it is taken as the relative path with respect to the current working directory.

Referring to HOME Directory

As mentioned earlier, each user on the Linux system is assigned a directory to store files. After logging in, a user is directly taken to the user's HOME directory. In other words, the working directory of the user after logging in to the Linux system is his or her HOME directory. While working on Linux, you may often feel the need to

Linux user and file management

access the HOME directory in a convenient way. In Linux, you can specify the path for your HOME directory with the tilde (~) sign. For instance, if Odroho wants to access the *nov13* file, he can specify the file name with the path as *~/Cricket/nov13* from anywhere in the file system.

2.5. Types of files

In Linux, all information is treated as a file. So, besides a user's program files and data files, there are also special files, such as those that contain information about directory contents or the various input/output devices connected to the system. In Linux, a device is also treated as a file and all the information going to, say, the VDU (Visual Display Unit) is treated as if it were being sent to a file. In Linux, there are three categories of files:

- Ordinary files
- Directory files
- Special files

Ordinary files

All the files created by a user come under this category of files. These include all the data files, program files, object files, and executable files. A user can make changes to such files.

Directory files

Linux automatically creates a directory file when a directory is created. This file has the same name as the directory, and contains information about the files under the directory. For example, for the directory, */home/Odroho*, there will be a directory file called *Odroho* in the directory, */home*, which contains information on all the files and directories under the directory, *Odroho*. A directory file cannot be modified by a user but is, instead, modified automatically by the system when a new file or a sub-directory is added to the directory.

Special Files

Most of the system files in Linux are special files. Special files are typically associated with input/output devices and are found in the standard Linux directories, such as *fdev* and *fetc*. Users cannot alter special files.

Operating System

2.6. Exercises

2.6.1. Multiple choice questions

- a. All the files are stored on the disk under one main directory called the
- (i) boot
 - (ii) bin
 - (iii) home
 - (iii) root
- b. The /dev directory
- (i) Contains all the HOME directories of users.
 - (ii) Has information specific to different utilities of Linux.
 - (iii) Stores all the device-related files for the system.
 - (iv) Contains libraries of data for compiler installed in the system.
- c. There are
- (i) Three types of files in Linux
 - (ii) Four types of files in Linux
 - (iii) Two types of files in Linux
 - (iv) Six types of files in Linux

2.6.2. Questions for short answers

- a. What are the files stored in /dev directory?
- b. Narrate the file naming convention of Linux.

2.6.3. Analytical questions

- a. Describe different directories under the '/' directory.
- b. Discuss Ordinary and Directory files of Linux.
- c. Describe the conception of relative path name with block diagram.

Lesson 3 : File System and Directory Structure

3.1. Learning Objectives

On completion of this lesson you will be able to know:

- ❖ About file access permission in Linux
- ❖ How to change file access permission

3.2. Determining the FAPs for a file

File Access Permissions (FAPs) refer to the permissions associated with a file with respect to the following:

Determining the FAPs for a file

- The file owner
- The group owner
- Other users

The following table summarizes the access permissions available for files and directories.

Access type	Denoted by	Action permitted on a file	Action permitted on a directory
Read	r	Allows you to display, copy, and compile the file	Allows you to list the contents of the directory
Write	w	Allows you to edit, rename or move the file to another location	Allows you to create new files and sub-directories within this directory
Execute	x	Allows you to execute the file provided the file also has the read permission	Allows you to move to that directory using the cd command.

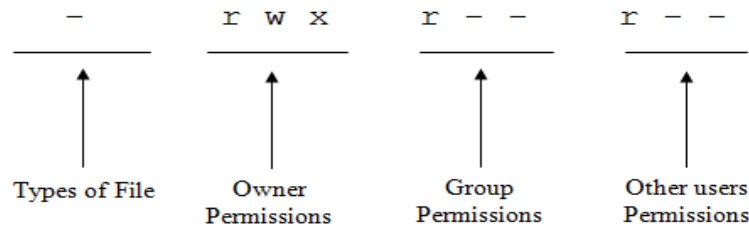
File Access Permissions.

The FAPs for a file can be viewed using the `ls -l` command.

[Steve@localhost Steve]\$ `ls -l`

```
Total 21
-rw-rw-r-- 1 Odroho Odroho 134  Nov 23 11:49 x.c
drwxr-xr-x 5 Odroho nazrul 1024  Nov 22 24:31 FAP
drwx----- 2 Odroho root 29024  Nov 24 10:01 Mail
-rwxr-xr-x 1 Odroho oyonti 1024  Nov 22 24:31 prog.txt
↑
FAP
```

Operating System



The permissions are displayed in the first column from the second position onwards.

The first three characters show the read, write, and execute permissions (in that order) for the file owner, the next three are the permissions for the group owner, and the last three are the permissions for other users.

If the permission is available, r, w, or x is displayed; else a hyphen is displayed.

3.3. Changing FAPs

The access permissions associated with a file or directory can be changed using the **chmod** command. Only the owner of a file can change the permissions associated with it.

Syntax

chmod mode file/s

You can specify the mode (permission for a type of user) in symbols or by using numbers (the absolute way). The symbolic format uses letters to represent permissions and the absolute way uses the numeric format.

Symbolic Mode

In the symbolic mode, the permission and the type of users for whom the permission is to be granted are given in symbols.

The table below states the symbols for different permissions:

Symbol	Meaning
r	Read
w	Write
x	Execute

Symbols for Permissions

The following table states the symbolic representation of entities.

Symbol	Meaning
u	Owner of the file or directory
g	Members of the same group
o	All the other users
a	All users

File Access Permissions

Linux user and file management

Linux allows you to change the FAPs for a specific user-type.

Example

```
[Odroho@localhost Odroho]$ ls -l x.c  
-rw-rw-r-- 1 Odroho IRDTECH 50 nov 24 22:02 x.c  
[Odroho@localhost Odroho]$ chmod u+x x.c
```

Here, 'u' indicates the file owner, '+' indicates that the permission is to be given, 'x' indicates the execute permission, and x,c is the file name.

3.4. Exercises

3.4.1. Multiple choice questions

- a. Access Permission 'w' allows you
- (i) To display copy and compile the file.
 - (ii) To edit, rename or move the file to another location.
 - (iii) To execute the file provided the file also has the read permission.
 - (iv) Allows you to move that directory using the cd command.
- b. Syntax of changing FAPs is
- (i) `chmod mode file/s`
 - (ii) `usermod mode file/s`
 - (iii) `chmod mode file/s`
 - (iv) `lsmod mode file/s`

3.4.2. Questions for short answers

- a. Define file access permission.
- b. How do you change FAPs of a file? Give syntax.

3.4.3. Analytical question

- a. Discuss in detail the changing procedure of FAPs.