Experiment No. 2

Title: Understanding File System of Linux
Implementation of Basic Linux Commands

Batch: B2 **Roll No: 16010420117** **Experiment No: 2**

Aim: Understanding Linux File system and executing basic commands in Linux

---

Resources needed: Any Open Source OS/CoCalc Linux terminal online

---

Theory:

Pre lab/Prior concepts:

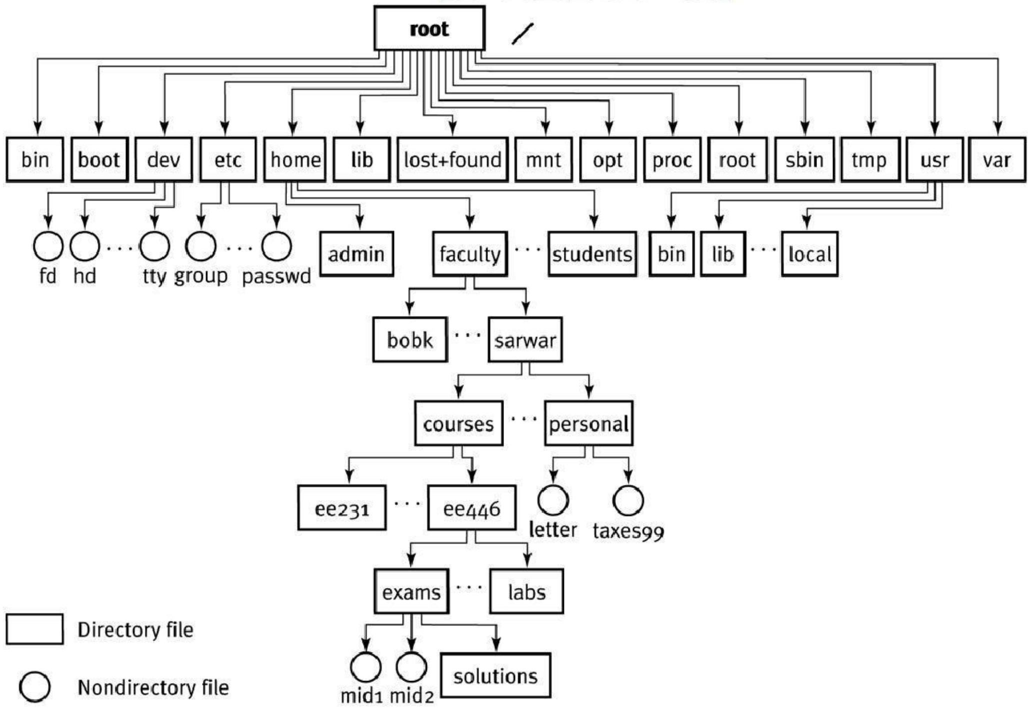
File -system management:

The file-system is the way an operating system manages all the files to be stored on the external storage - binaries, images, etc. Linux uses an advanced version of the Extended File -system from Unix, called ext2.

An important feature of the ext2 file-system is that it treats everything as files - directories are also represented as files containing pointers to other files. This does not stop only at directories, even the hardware can be addressed as files under the standard directory /dev.

Directories

In Linux, different partitions need not be assigned special names to access them. This is because Linux uses a standard directory structure to take care of all partitions. This structure ensures that a particular file for a particular program will almost always be present at the same place on any machine running Linux.



ne drawback in Linux has been the susceptibility of ext2 to damages due to power failures. But there has been significant development to rectify this shortcoming through the introduction of journaling file-systems like reiserfs, ext3. A journaling file-system has the inherent capability to be recovered to a stable state in case of crashed due to power failures.

The figure above show the hierarchical organization of directories created on the filesystem by default, and which are considered to be standard for all installations. Let us take a look at some of the more important ones.

 /dev is the directory through which all the devices on the machine are accessible as

 files. These include the serial terminals (COM ports), modem, mouse, sound card   ... everything!

 /var contains most of the "variable" data such as mails, log files, databases, etc.

 /usr is where almost all the packages get installed. Discounting /mnt, it is the largest directory on the hard-disk. This directory itself has a pretty complicated sub-directory system used by the packages.

 /etc contains all the configuration files, used by the operating system itself as well

      as various packages installed on the system.

 /home contains the home directories for all the users created on the system.

 /mnt is conventionally used as the base for all directories which are not part of the standard directory hierarchy. It is most commonly used to mount the cd-rom drives, floppy drives, non-Linux partitions, etc.

 /root is the home directory of the privileged user or the system administrator, called "root".

In Linux there are different commands available to perform file operations like create, edit, copy, rename and move file.

We can assign read, write, and execute permissions to file. Every file in Linux has a list of permissions attached to it, that specifies the kind of access that different users have to that file. The file permissions are also called its access modes.

The very first column in the above output defines the access modes for the file. Their meaning will become apparent when we look at the way access modes are represented. Thes e can be expressed as either octal numbers or mnemonics.

Linux commands for file handling:

File handling commands include commands for creating files/directories, navigating through file system.

Commands to create a file

touch: creates one or more empty file(s).

syntax:

touch filename1 [filename2] [filename3]

cat: creates/displays file

syntax:

cat [options] [filename]

cat > [filename] → To create a file

cat [filename] → To view a file

cat >> [filename] → To append text to a file

mkdir : Make directory (or) To create a directory

Syntax :

```
mkdir [options] [dirname]
```

mkdir –p /home/abc   creates directory with subdirectories

mkdir –m 777 ~/data   creates data directory in home and assigns read, write, execute permissions to it

rmdir: Remove an empty directory. If you want to remove a directory with files in it type " rm -R directory"

Syntax:

```
rmdir dirname
```

This will only remove directory if it's empty otherwise it will exit with an error message.

pwd: Print working directory. Print the absolute (complete) path to the directory the user is currently in.   **Syntax**: pwd

This will tell you the full path to the directory you are in, for example it may output "/usr/local/bin" if you are currently in that directory.

cd: Change directory. Use " cd .." to go up one directory.

One dot '.' represents the current directory while two dots '..' represent the parent directory. " cd -" will return you to the previous directory (a bit like an "undo").You can also use cd absolute path or cd relative path

ls : List the Files and Directories within the current directory.

Syntax:

```
ls              [options]
```

Options:

-l List file with permission.

a List hidden file.

-i   List files and inode number.

You can use ls -d to show directories that match an exact string, or use standard wildcards. Type " ls -d */" to list all subdirectories of the current directory. Depending on the setup of your aliases you may simply be able to type lsd as the equivalent to ls -d */ .

Examples for ls -d :

ls -d

Lists all subdirectories of current directory.

-R List directories , sub directories and their contents.

```
ls  -options string
```

This lists files using a certain string. The string can contain standard wildcards to list multiple files

```
ls -d string*
```

Lists directories that start with "string".

```
ls -d /usr/*/*/doc
```

Lists all directories that are two levels below the /usr/ directory

chmod: change mode(permissions) for file/directory

```
Chmod [rwxrwxrwx]/[421412421] filename
```

rm: Remove/delete a file(s) or directories(s). You can use standard wildcards with this command

Syntax:

```
rm -options file_or_folder
```

You can of course use standard wildcards to delete multiple files or multiple directories and files.

Use the -R or -r option to remove recursively, this removes everything within subdirectories. Also try the -f option to force removal (useful when you don't want to be prompted).

mv: Move a file or a directory to a new location or rename a file/directory.
Rename example:

mv filename1 filename2 Renames
filename1 to filename2.

To move a file or directory, simply
type: mv original_file_or_folder new
location

Note that this command can use standard wildcards to move files (not for renaming).

Move and rename

Note that you can also move and rename a file in a single command. The difference is with the destination (right hand side) you change the filename to the new name of the file.

For example typing: mv /etc/configuration.txt
/home/joe/backupconfig
This would move the file "configuration.txt" to /home/joe/ and rename it "backupconfig"

cp: Copy a file. Has a number of useful options, such as -R (or -r) which recursively copies director ies and subdirectories.
Syntax: cp -options file_or_files
new_location

Examples:
cp file1 file2
Simply copy file1 to file2 (in the same directory).

cp /tmp/file1 ~/file2 /mnt/win_c

find: The following examples illustrate typical uses of the command find for finding files on a computer.

find / -name game

Looks for a file named "game" starting at the root directory (searching all directories including mounted filesystems). The `-name' option makes the search case sensitive.
You can use the `-iname' option to find something regardless of case. find
/home -user joe
Find every file under the directory /home owned by the user joe

There are other commands to locate file like locate, whereis, which, whatis etc.

User administration:

Linux is a true multiuser environment. This means that the system can support different users with different privileges. Each user has access to a predefined set of system services and his/her own private data. The private data in turn can be shared with other users by granting access privileges to them. All the privilege checking is done with the help of user accounts maintained by the system. Linux provides a number of commands that can be used to create and manage user accounts. These are supported by a number of files and directories under /etc that are used to hold information about the users. The facilities provided can be broadly classified into two categories - user management and group management.

User accounts

Traditionally, the information regarding the users is placed in a file called /etc/passwd. This contains the login name, full name, home directory and other info in a standard format. It may also contain the encrypted password used by the user, hence the name of the file. But nowadays, better ways for user authentication are used, which store the password elsewhere.

Superuser Account

By default, every Linux installation has a specially privileged account called the root or superuser. This user has complete access to all the services and resources present on the system. The account is normally owned by the system administrator, and used to carry out special task that require special privileges not available to normal accounts. A person who logs in as root can modify any file on the entire system irrespective of the actual owner of the file and run any program anywhere on the system. As such, it is the most powerful account and has to be used with caution. Mistakes made while logged in as root can prove very dangerous to the system; hence its extremely important that the user should avoid using the root account unless absolutely necessary.

All other accounts are said to be non-privileged, since they have only access to a limited amount of services. Their privileges can be further controlled by use of groups which are used for collective management of user accounts.

The concept of ``groups'' in Linux

In Linux, users are divided into logical collections called groups. These are used to confer various kinds of privileges to system objects to a group of users together. One user may belong to a number of groups, but he/she will always have a default group, along with other groups which are said to be supplementary. When a user is added to a particular group, all the privileges that are conferred on the group are also conferred on the user.

File group

In order to implement access priviges, the first thing to do is define an owner and a group for the file. This information establishes who can claim control of the file. A file is given an owner and a group as soon as it is created. Usually owner is the current user and the group is the group of the directory within which the file is created; but this is system dependent.

Linux commands for user administration:

Adding new users

The standard command useradd can be used to create a new user on the system. It is one of a family of commands for user management, that can only be invoked by a user with special privileges, ie, the root. The most common arguments provided to the command are as follows:

useradd :

To create a new user account and login directory for that new account. Useradd will create new entries in system files.

Syntax useradd

[*options*] [*user*]

Options

-c Comment field.

-d Home directory

-e Account expiration date.

To assign password to this user use following command

passwd [password]

When a new user account is created, its entries update the following system files.

1. /etc/passwd
2. /etc/group
3. /etc/shadow

Removing users

The command userdel is used to remove an existing user from a system. This can be invoked as follows: userdel -r <username>

This command modifies all the system files, deleting all information about the specified username. The -r option will cause the user's home directory to be deleted along with any files or sub-directories it may contain.

Modifying user information

The command usermod is used to modify information about an existing user. The arguments accepted by this are almost the same as those accepted by useradd command. The only difference is that it modifies existing entries rather than creating new one's.

```
usermod –G newgroupname username
```

Managing groups

The superuser can use a set of commands analogous to the once used for user management.

New group is created by hand -editing the file /etc/group or by using groupadd command.

Syntax:

```
groupadd groupname groupdel
```

command is used to remove the group.

Syntax:

```
groupdel  groupname
```

Groupmod is used to rename the existing groupname.

Syntax:

```
groupmod –n newname oldname
```

These commands have functions similar to the corresponding one's used for user accounts. Again, every group is assigned a unique number called GID, ie, the Group ID.

chgrp: change group of file **Syntax:**

chgrp newgroupname filename

chown: change owner of file **Syntax:** chown user:group filename

Changing Ownership and Group

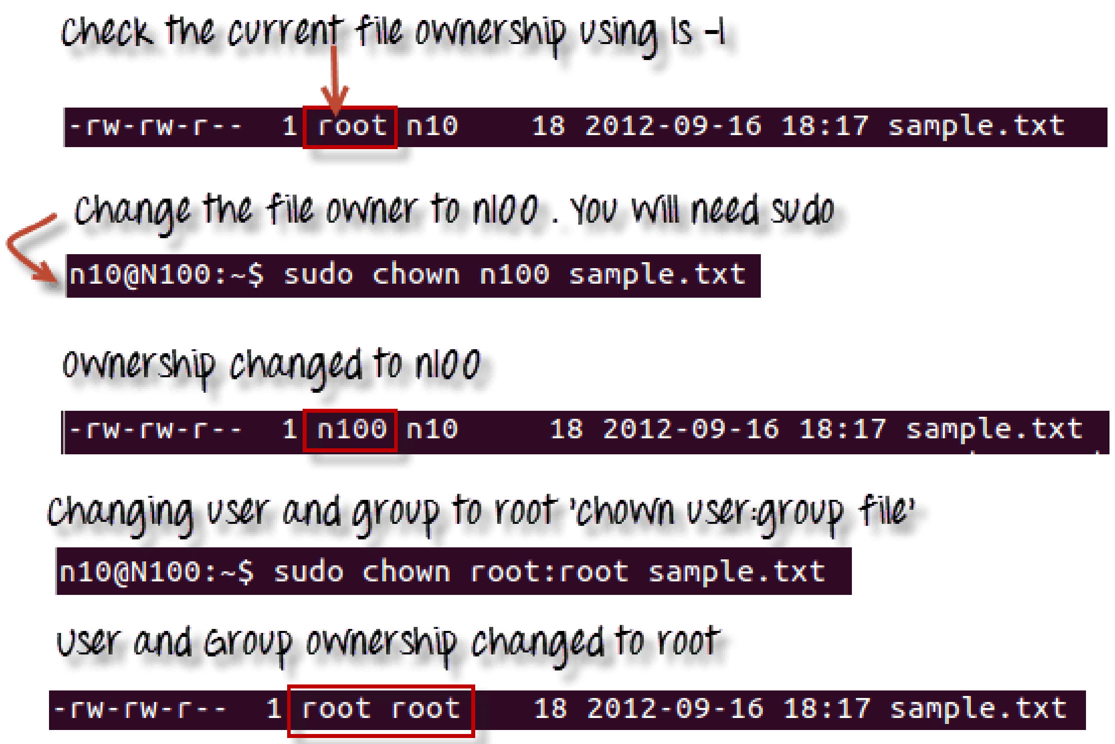
For changing the ownership of a file/directory, you can use the following command:

chown user

In case you want to change the user as well as group for a file or directory use the command

chown user:group filename

Let's see this in action



Check the current file ownership using ls -l

```
-rw-rw-r--  1 root n10    18 2012-09-16 18:17 sample.txt
```

Change the file owner to n100 . You will need sudo

```
n10@N100:~$ sudo chown n100 sample.txt
```

ownership changed to n100

```
-rw-rw-r--  1 n100 n10     18 2012-09-16 18:17 sample.txt
```

Changing user and group to root 'chown user:group file'

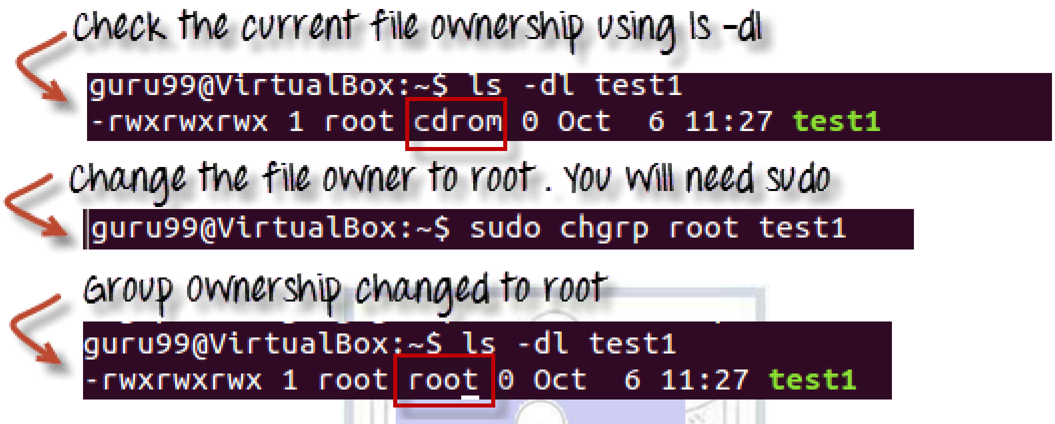
```
n10@N100:~$ sudo chown root:root sample.txt
```

User and Group ownership changed to root

```
-rw-rw-r--  1 root root    18 2012-09-16 18:17 sample.txt
```

In case you want to change group-owner only, use the command ***chgrp group_name filename***

'**chgrp'** stands for change group.



Check the current file ownership using ls -dl

```
guru99@VirtualBox:~$ ls -dl test1
-rwxrwxrwx 1 root cdrom 0 Oct  6 11:27 test1
```

Change the file owner to root . You will need sudo

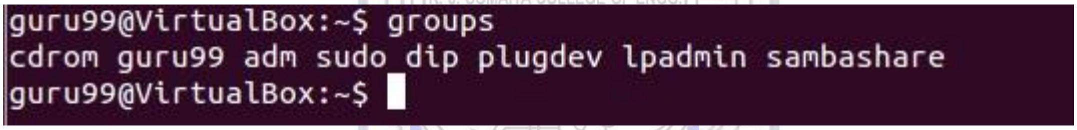
```
guru99@VirtualBox:~$ sudo chgrp root test1
```

Group ownership changed to root

```
guru99@VirtualBox:~$ ls -dl test1
-rwxrwxrwx 1 root root 0 Oct  6 11:27 test1
```

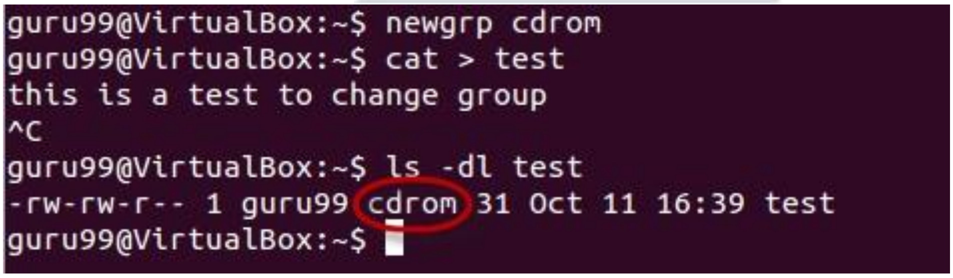
Note:

- The file /etc/group contains all the groups defined in the system
- You can use the command "groups" to find all the groups you are a member of



```
guru99@VirtualBox:~$ groups
cdrom guru99 adm sudo dip plugdev lpadmin sambashare
guru99@VirtualBox:~$ 
```

- You can use the command newgrp to work as a member a group other than your default group



```
guru99@VirtualBox:~$ newgrp cdrom
guru99@VirtualBox:~$ cat > test
this is a test to change group
^C
guru99@VirtualBox:~$ ls -dl test
-rw-rw-r-- 1 guru99 cdrom 31 Oct 11 16:39 test
guru99@VirtualBox:~$ 
```

- You cannot have 2 groups owning the same file.
- You do not have nested groups in Linux. One group cannot be sub-group of other
- x- eXecuting a directory means Being allowed to "enter" a dir and gain possible access to sub-dirs

chage: command let you specify an expiration limit for a user's account and password .

Syntax: chage
[option] username

Option
- l lists the current password expiration.

-m set the min. days to change the password. M
set the max. days to change the password. E
specific expiration date for user account.
-I set inactive period (in days)
-W warning period, number of days before expiration.

Process management:

Process is running instance of a program. Process manangement starts with init command which is present in a file named /etc/initta. Init is the first process that runs on the system and can start other processes depending upon the run level. Every process has process idenfication(PID) number. PID of init is 1.

Different types of process:

Daemon process

User process

To display currently running processesdon linux system following **ps** command is used. Output of the ps command has following fields

Column	HeaderContents
%CPU	How much of the CPU the process is using
%MEM	How much memory the process is using
ADDR	Memory address of the process
C or CP	CPU usage and scheduling information
NI	nice value
PID	Process ID number
PPID	ID number of the process's parent process
PRI	Priority of the process
RSS	Real memory usage
S or STAT	Process status code
START or STIME	Time when the process started
SZ	Virtual memory usage
TIME	Total CPU usage
TT or TTY	Terminal associated with the process
UID or USER	Username of the process's owner
WCHAN	Memory address of the event the process is waiting for

To kill process using PID command **kill** is used. And to kill process by name **killall** command is used   Job management:

Job management uses the concept of foreground and background processes. Foreground process always have access to the standard input stream to receive command from the user and the standard output stream to print information to the user. Where the programs which run in the background are do not have access to these streams. So can’t interact with user .

Compiling and building software related programs can run in background. Through command we can make any program executing in background to run in foreground and vice versa.

Linux commands for Process Management:

ps: used to list the process.To display a information about process specific to the active terminal.

Syntax:

ps

[option]

Options available:

- a :   To display all process, excluding process not controlled by a terminal

-x : To include all process not controlled by a terminal, such as daemon process. -l : To long list including more information such as process owner's user id.

-u : To display user name of the process owner. -e
: To view the all process id

kill: used to send a particular process to get clear.

Syntax :

kill -9 <process id>

ps - aux is the command to view all process name and id

Example: kill -9 3743 (This example shows the kill command, option and the process id to kill.)

Renice: used to set the priority of a process.The priority value can range from -20 (highest priority) to 19 (Lowest priority) **Syntax :**

renice - +(minus symbol or plus symbol) <priority number> -p <process id> Example : renice -15 -p 1970

jobs command helps you to view the background processes.

Fg command is used to run the background process in the foreground (Front End) jobs <enter> to view the process running and job numbers fg %2 (job number)

bg command is used to resume the suspended command in the back grounded.

Activities: Activities:

1. Try out different file management commands. Browse manual pages of the commands and try-out different options of the commands
2. User Administration

Create a two users and assign passwords □

Check for creation of login directory of user account. □

□

See and note down userid, groupid of this user from /etc/passwd file and group

□ form /etc/group file □

Delete any one of the two users created above □

Create a new group1 and add user to this group □

□ Create three users and add them to a new group2. □

Create a directory and use chgrp to change group of this directory to group2 □

□

Now login using first user of group 1and try accessing this directory of group2 note down output □

□

Check group and owner of the file and then change both group and owner of that

□ file using chown □

Explore chage command □

3. process management

- explore ps command □
- list all jobs □
- start any new process in background □
- start its execution in foreground □
- suspend execution of this process
- resume its execution in background

Results:

Students should log the results of these commands in a separated file (Do not use write-ups to log your results) and upload it on the drive shared by the faculty. Use *.txt/.doc/.docx extensions only while saving the file. Do not take snapshots.

This file must contain on the top:
Name:
Roll No.
Exp No.
Batch: Date:

```
kjsce@ubuntu:~$ touch new.txt
kjsce@ubuntu:~$ cat>new.txt
hi, i am aayush
^C
kjsce@ubuntu:~$ cat new.txt
hi, i am aayush
kjsce@ubuntu:~$ cat>>new.txt
hi everyone
^C
kjsce@ubuntu:~$ cat new.txt
hi, i am aayush
hi everyone
kjsce@ubuntu:~$ mkdir abc
kjsce@ubuntu:~$ mkdir -m 777 kmmm
kjsce@ubuntu:~$ ls -l
drwxrwxrwx  2 kjsce kjsce      4096 Sep  12 11:54 kmmm
kjsce@ubuntu:~$ pwd
/home/kjsce
kjsce@ubuntu:~$ cd abc
kjsce@ubuntu:~/abc$ ls -d
.
kjsce@ubuntu:~/abc$ cat>test.txt
apple
^C
kjsce@ubuntu:~/abc$ cat>test1.txt
^C
kjsce@ubuntu:~/abc$ ls -d
.
kjsce@ubuntu:~/abc$ mv test1.txt test2.txt
kjsce@ubuntu:~/abc$ cp test.txt test2.txt
kjsce@ubuntu:~/abc$ cat>>test2.txt
^C
kjsce@ubuntu:~/abc$ sudo su
[sudo] password for kjsce:
root@ubuntu:/home/kjsce/abc# useradd ss
root@ubuntu:/home/kjsce/abc# useradd as
root@ubuntu:/home/kjsce/abc# passwd as
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@ubuntu:/home/kjsce/abc# groupadd as66
root@ubuntu:/home/kjsce/abc# groupmod -n soham66 as66
root@ubuntu:/home/kjsce/abc# adduser bhoirsoham
Adding user `bhoirsoham' ...
Adding new group `bhoirsoham' (1005) ...
Adding new user `bhoirsoham' (1004) with group `bhoirsoham' ...
Creating home directory `/home/bhoirsoham' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bhoirsoham
```

```
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@ubuntu:/home/kjsce/abc# userdel -r bhoirsoham
userdel: bhoirsoham mail spool (/var/mail/bhoirsoham) not found
root@ubuntu:/home/kjsce/abc# ps -a
  PID TTY          TIME CMD
 3022 pts/4    00:00:00 sudo
 3025 pts/4    00:00:00 su
 3026 pts/4    00:00:00 bash
 3195 pts/4    00:00:00 ps
root@ubuntu:/home/kjsce/abc# ps -x
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init splash
    2 ?        S      0:00 [kthreadd]
    4 ?        I<     0:00 [kworker/0:0H]
    6 ?        I<     0:00 [mm_percpu_wq]
    7 ?        S      0:00 [ksoftirqd/0]
    8 ?        I      0:01 [rcu_sched]
    9 ?        I      0:00 [rcu_bh]
   10 ?        S      0:00 [migration/0]
   11 ?        S      0:00 [watchdog/0]
   12 ?        S      0:00 [cpuhp/0]
   13 ?        S      0:00 [cpuhp/1]
   14 ?        S      0:00 [watchdog/1]
   15 ?        S      0:00 [migration/1]
   16 ?        S      0:00 [ksoftirqd/1]
   18 ?        I<     0:00 [kworker/1:0H]
   19 ?        S      0:00 [cpuhp/2]
   20 ?        S      0:00 [watchdog/2]
   21 ?        S      0:00 [migration/2]
   22 ?        S      0:00 [ksoftirqd/2]
   23 ?        I      0:00 [kworker/2:0]
   24 ?        I<     0:00 [kworker/2:0H]
   25 ?        S      0:00 [cpuhp/3]
   26 ?        S      0:00 [watchdog/3]
   27 ?        S      0:00 [migration/3]
   28 ?        S      0:00 [ksoftirqd/3]
   30 ?        I<     0:00 [kworker/3:0H]
   31 ?        S      0:00 [kdevtmpfs]
   32 ?        I<     0:00 [netns]
   33 ?        S      0:00 [rcu_tasks_kthre]
   34 ?        S      0:00 [kauditd]
   37 ?        S      0:00 [khungtaskd]
   38 ?        S      0:00 [oom_reaper]
   39 ?        I<     0:00 [writeback]
   40 ?        S      0:00 [kcompactd0]
   41 ?        SN     0:00 [ksmd]
   42 ?        SN     0:00 [khugepaged]
   43 ?        I<     0:00 [crypto]
   44 ?        I<     0:00 [kintegrityd]
   45 ?        I<     0:00 [kblockd]
   47 ?        I      0:00 [kworker/3:1]
   48 ?        I<     0:00 [ata_sff]
   49 ?        I<     0:00 [md]
   50 ?        I<     0:00 [edac-poller]
   51 ?        I<     0:00 [devfreq_wq]
   52 ?        I<     0:00 [watchdogd]
   55 ?        S      0:00 [kswapd0]
   56 ?        I<     0:00 [kworker/u9:0]
   57 ?        S      0:00 [ecryptfs-kthrea]
   99 ?        I<     0:00 [kthrotld]
  100 ?        I<     0:00 [acpi_thermal_pm]
  105 ?        I<     0:00 [ipv6_addrconf]
  114 ?        I<     0:00 [kstrp]
  131 ?        I<     0:00 [charger_manager]
  173 ?        S      0:00 [scsi_eh_0]
  174 ?        I<     0:00 [scsi_tmf_0]
  175 ?        S      0:00 [scsi_eh_1]
  176 ?        I<     0:00 [scsi_tmf_1]
  177 ?        S      0:00 [scsi_eh_2]
  178 ?        I<     0:00 [scsi_tmf_2]
  179 ?        S      0:00 [scsi_eh_3]
  180 ?        I<     0:00 [scsi_tmf_3]
  182 ?        I      0:00 [kworker/u8:3]
  184 ?        S      0:00 [i915/signal:0]
  185 ?        S      0:00 [i915/signal:1]
  186 ?        S      0:00 [i915/signal:2]
  196 ?        I<     0:00 [kworker/3:1H]
  197 ?        S      0:00 [scsi_eh_4]
  198 ?        I<     0:00 [scsi_tmf_4]
  199 ?        S      0:00 [usb-storage]
  226 ?        Ss     0:03 mount.ntfs /dev/sda4 /root
  235 ?        S<     0:01 [loop0]
  238 ?        I<     0:00 [kworker/1:1H]
  239 ?        I<     0:00 [kworker/2:1H]
  247 ?        I<     0:00 [kworker/0:1H]
  248 ?        S      0:00 [jbd2/loop0-8]
  249 ?        I<     0:00 [ext4-rsv-conver]
  290 ?        Ss     0:00 /lib/systemd/systemd-journald
  336 ?        Ss     0:00 /lib/systemd/systemd-udevd
  359 ?        S<     0:00 [loop1]
  362 ?        S<     0:00 [loop2]
  627 ?        S      0:00 [irq/27-mei_me]
  917 ?        Ss     0:00 /usr/sbin/acpid
  921 ?        Ss     0:00 /lib/systemd/systemd-logind
  931 ?        Ss     0:00 /usr/sbin/cron -f
  956 ?        Ssl    0:00 /usr/sbin/NetworkManager --no-daemon
  957 ?        Ssl    0:00 /usr/sbin/thermald --no-daemon --dbus-enable
  960 ?        Ssl    0:00 /usr/lib/accountsservice/accounts-daemon
```

```
  961 ?        Ssl    0:00 /usr/sbin/ModemManager
  964 ?        Ssl    0:00 /usr/lib/snapd/snapd
  998 ?        Ss     0:00 /usr/sbin/irqbalance --pid=/var/run/irqbalance.pid
 1008 ?        Ssl    0:00 /usr/lib/policykit-1/polkitd --no-debug
 1026 ?        SLsl   0:00 /usr/sbin/lightdm
 1046 tty7     Ssl+   0:40 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/ru
 1051 ?        Ssl    0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unatt
 1056 ?        Ss     0:00 /usr/sbin/sshd -D
 1181 ?        Sl     0:00 lightdm --session-child 12 19
 1306 ?        Ssl    0:00 /usr/lib/upower/upowerd
 1588 tty1     Ss+    0:00 /sbin/agetty --noclear tty1 linux
 1942 ?        Ssl    0:00 /usr/lib/udisks2/udisksd --no-debug
 1988 ?        Ssl    0:00 /usr/lib/x86_64-linux-gnu/fwupd/fwupd
 2518 ?        Ss     0:00 /usr/sbin/cupsd -l
 2519 ?        Ssl    0:00 /usr/sbin/cups-browsed
 2526 ?        I      0:00 [kworker/1:2]
 2744 ?        I      0:00 [kworker/1:0]
 2748 ?        I      0:00 [kworker/2:1]
 2751 ?        I      0:00 [kworker/3:0]
 2770 ?        I      0:00 [kworker/0:1]
 2856 ?        I      0:00 [kworker/0:0]
 2965 ?        I      0:00 [kworker/u8:1]
 3005 ?        R      0:00 [kworker/u8:0]
 3022 pts/4    S      0:00 sudo su
 3025 pts/4    S      0:00 su
 3026 pts/4    S      0:00 bash
 3196 pts/4    R+     0:00 ps -x
root@ubuntu:/home/kjsce/abc# ps -u
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1046  1.0  1.1 432508 96620 tty7     Ssl+ 11:22   0:40 /usr/lib/xorg/X
root      1588  0.0  0.0  15936  1788 tty1     Ss+  11:22   0:00 /sbin/agetty --
root      3022  0.0  0.0  54792  3880 pts/4    S    12:15   0:00 sudo su
root      3025  0.0  0.0  54376  3584 pts/4    S    12:15   0:00 su
root      3026  0.0  0.0  21224  3824 pts/4    S    12:15   0:00 bash
root      3197  0.0  0.0  37364  3316 pts/4    R+   12:26   0:00 ps -u
```

Outcomes: CO1: Understand basic structure of modern operating system
CO4: Demonstrate open source standards usage

Conclusion:
Students must have basic understanding of Linux Operating System.

Grade: AA/AB/BB/BC/CC/CD/DD

Signature of faculty in-charge with date

References:
Books/ Journals/ Websites:

1. Richard nd Edition Blum edition, and Christine Wiley, 2012 Bresnahan, “Linux. Command Line & Shell Scripting”, II