Experiment 2 : Familiarization of Linux Environment - Linux Basic Commands

What Is Linux?

Linux is an operating system's kernel. You might have heard of UNIX. Well, Linux is a UNIX clone. But it was actually created by Linus Torvalds from Scratch. Linux is free and open-source, that means that you can simply change anything in Linux and redistribute it in your own name! There are several Linux Distributions, commonly called “distros”.
Ubuntu Linux
Red Hat Enterprise Linux
Linux Mint
Debian
Fedora

Linux is Mainly used in servers. About 90% of the internet is powered by Linux servers. This is because Linux is fast, secure, and free! The OS that runs in about 80% of the smartphones in the world, Android, is also made from the Linux kernel.

Linux Shell or “Terminal”

So, basically, a shell is a program that receives commands from the user and gives it to the OS to process, and it shows the output. Linux's shell is its main part. Its distros come in GUI (graphical user interface), but basically, Linux has a CLI (**command line interface**).
To open the terminal, press **Ctrl+Alt+T in Ubuntu**, or press Alt+F2, type in gnome-terminal, and press enter.

Lesson1 - Basic commands to navigate directories

`pwd`

will display your current working directory.Yeah,your home is a directory. Now let's try to create a new directory.type the following on the prompt

`mkdir dir1`

and press the enter key.

Wow,now you created a new directory. Let's say you want to create more than one directory instead of invoking mkdir multiple(three) times-like.

```
mkdir dir2
mkdir dir2/dir3
mkdir dir2/dir3/dir4
```

you can simply use

```
mkdir -p dir2/dir3/dir4
```

"-p" option will create parent directories for "dir4" as needed. In this case,it creates dir2,dir3 automatically.Now we have created 4 directories.How to view them?

To view type 'ls' and press enter

```
ls
```

listed `dir1 dir2` as directory content right? That's exactly what we wanted

Where is the missing dir3,dir4? They are created inside dir2. They won't be listed with simple commands like `ls`. you need to use the "complex" command to view them. Try this:

```
ls -R
```

btw -R stands for recursive.

Okay,we have created new directories and listed them.Now lets move into a new directory.

```
cd dir2
```

cool,you have changed to dir2 Now confirm this location by using previously learned `pwd` command.

```
pwd
```

To move into next directory dir3

```
cd dir3
```

will place you under the "dir3" directory.

```
pwd
```

`Tips and tricks:` Typing

```
cd ..
```

will move to the parent directory.i.e dir2.

```
pwd
```

Now type,

```
cd -
```

will move you to the previous working directory i.e dir3.

```
pwd
```

and a simple

```
cd
```

will move to your home directory.

```
pwd
```

That's it.You have successfully completed lesson1 Now to start the next lesson.

Lesson2 - Create files, display contents and stats

During Lesson1,you have learned how to create directories.

Lets learn to create a new file,

```
touch file1.txt
```

and press enter key

touch command will create a new file or change the timestamp of an existing file.

```
ls
```

Now try again,

```
touch file2.txt
```

will create an empty new file ,if the file does not already exist. to view directory contents ,you can also use

```
dir
```

`dir` is used to list directory contents.Yeah,as you guessed it correctly , `dir` is equivalent to `ls -C -b.`

That is, by default files are listed in columns, sorted vertically, and special characters are represented by backslash escape sequences. To clear a screen,the command is

```
clear
```

Viola! terminal screen is cleared!!! Lets print some message on the terminal,

```
echo "hello"
```

Cool! the message is displayed on the screen. Lets redirect the message to a new file instead of screen.

```
echo "hello" > hello.txt
```

To append data you must use >> not just >

```
echo "linux" >> hello.txt
echo "world" >> hello.txt
```

Done.

```
ls
```

To view the file content ,do

```
cat hello.txt
```

so now you have viewed the file content.`cat` is used to display the entire file content.

To view only first two lines from the file

```
head -2 hello.txt
```

see,it showed us the first two lines from files. By default,`head` will display the first 10 lines when you run,

```
head hello.txt
```

Now how to view last two lines?.It's simple,use `tail`

```
tail -2 hello.txt
```

cool. Thus `head` will be used to display lines from beginning and `tail` will be used to display the last few lines. As with `head`

```
tail hello.txt
```

by default will display the last 10 lines from the line.

Let's check some stats of the files and directories we have created so far.

```
stat hello.txt
```

carefully examine a few important fields of the output. The first line shows the `filename`.second line says its a `regular file` with size as `18`.Third line shows `Inode` number and no.of `links` to that inode.

Fourth one,says `owner(Uid),group(Gid)` who has read-write permission but other have read permission.Final three lines show `access,modified and change` time.They mean:

```
access - when the file was last accessed/read.
modified - when the contents was last
     modified written.
change - denotes changes to files metadata
    like changing user permission.
```

Now let's do a `stat` in the directory.

```
stat dir1
```

Compare the previous `stat` "hello.txt" output with "dir1",before you move. especially find out "dir1" type.That marks the end of lesson2!.Well done.

Lesson3 - Copy,rename,delete files

In Lesson1,you learned about directories. With `Lesson2`,you learned about files. Now let's learn general file operations.

Now check this command

```
du
```

It displays the disk usage of the current directory.(Please note the current total of du output).Use the h switch to output in a human readable format and the x switch to exclude other file systems and ~ denotes your home.

```
du -xh ~
```

Now let's copy `hello.txt` to the dir2 directory.

```
cp hello.txt dir2
```

now the file is copied to a new location.Now compute the usage again using, `du` now you should see usage has been increased by file size.

`Tips and tricks:`

```
cp hello.txt dir2/file2.txt
```

This will copy hello.txt into dir2 at the same time, rename it as "file2.txt".

```
cp -r dir2/*.txt dir2/dir3
```

This will copy all files ending with ".txt" from dir2 into dir2/dir3.

```
cp -r dir2/dir3  .
```

This will copy the directory named "dir3" to the current directory.

Use `ls`,it should show you dir3.

Now we have copied a few files,how do we verify its file integrity?simple `cat` should be enough.

now lets move to another command,

```
mv hello.txt dir2/dir3/dir4/hi.txt
```

will move a file into directory dir4 and name it as hi.txt. `So how is mv different from cp?`.Try `ls` it will not show hello.txt.

When you use cp there exists two copies of a file (similar to copy-paste "ctrl-c" and "ctrl-v") with mv there is one copy (its cut-paste ctrl-x and ctrl-v). Unlike (cp,rm) other commands mv don't need "-r" for directories.

create a new directory dir5

```
mkdir dir5
```

now

```
mv dir2/*.txt dir5
mv dir5  dir50
```

will move all "*.txt" files under dir2 into dir5. then rename the directory "dir5" as "dir50".

with mv command we moved hello.txt under dir4,instead of accessing them as dir2/dir3/dir4/hi.txt everytime,we can create a link and after that,you can access or edit `dir2/dir3/dir4/hi.txt` file as simply `hello`

```
ln  dir2/dir3/dir4/hi.txt hello
```

Great! you have created a link.

```
stat hello
```

and perform

```
stat dir2/dir3/dir4/hi.txt
```

see both uses same inode and link count shown as 2.

To remove individual file use

```
rm file2.txt
```

will prompt you with a message.`rm: remove regular empty file 'file2.txt'?` type `y` to delete the file.To remove directory, first remove it's contents using option "r",

```
rm -r dir50/*
```

```
Tips and tricks:
```

If you want to remove files content without being prompted for confirmation use -f option. It's extremely dangerous to use "rm -rf",because you may delete very important files by mistake-so make sure you delete correct files before running rm -rf"

```
rm -rf junk/*
rmdir  dir50
```

rmdir will remove an empty directory. so that's end of lesson3.

Lesson 4 : Compile and Run a C Program on Ubuntu Linux

Step 1. Open up a terminal

Open up a terminal by clicking the right button of mouse and selecting Open Terminal from it.

Step 2. Use a text editor to create the C source code

Type the following command

gedit helloworld.c

This will open up a text file helloworld with extension (.c)

Now enter the C source code shown below:

```
#include<stdio.h>
int main()
{
printf("Hello World");
}
```

After writing the code save the file and close the text editor.

Note: *The file will be by default stored in the home folder. It is also possible to store the file in any user defined directory also.*

Step 3. Compile the program.
Type the following command in Terminal window.

gcc helloworld.c -o helloworld

This command will invoke the GNU C compiler to compile the file helloworld.c and output (-o) the result to an executable called helloworld. This file will be present in the same directory as the program file.

Note: Suppose if we had saved the program file in some other directory than the home directory, then using cd command we need to browse to that location and then type the above command.

Step 4. Execute the program.
Type the command

./helloworld

This should result in the output

Thus, we have successfully completed the creation of a C program and its execution in Linux Ubuntu Linux.