OPERATING SYSTEMS – INTRODUCTION TO LINUX

OCTOBER 7, 2016

16CSCI03I - OPERATION SYSTEMS

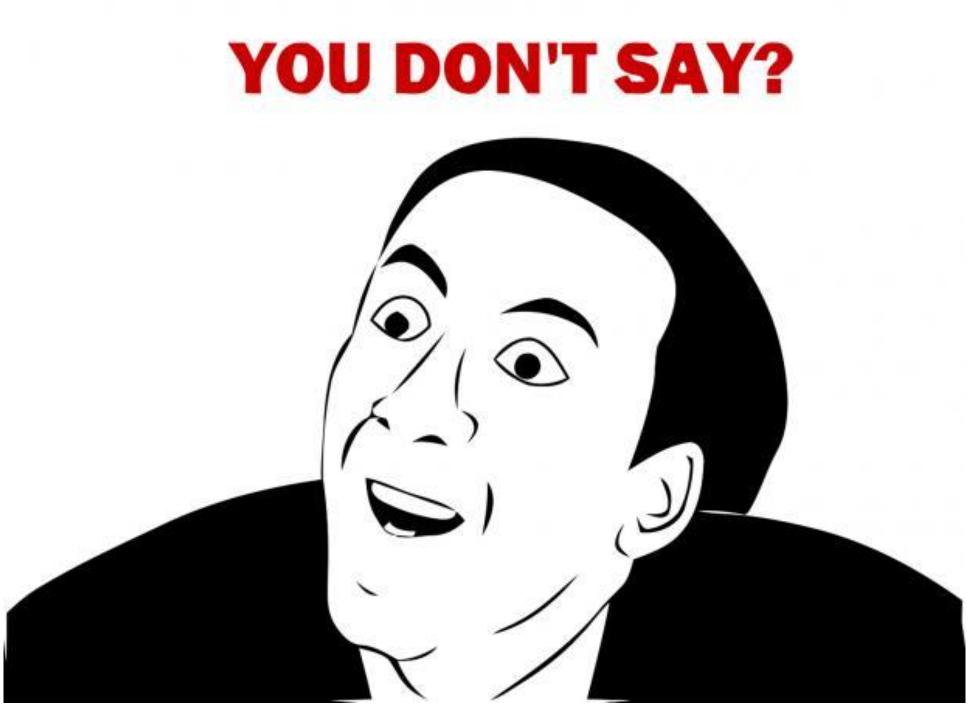
[Company address]

Operating Systems – Introduction to Linux

Introduction

What is Linux?

Linux is an operating system.



Again. What is Linux?

UNIX is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.

Linux Pros:

1. Linux is free; most of other operating systems are paid. Microsoft Windows is a paid operating system.
2. Linux is portable to any hardware platform.

3. Linux is secure and versatile. Surprise Surprise! You don't need an anti-virus installed on your computer if you are running Linux. Learn more at https://goo.gl/kt0DNE (this is optional to read, but worth it!)

4. Linux is scalable, it can startup and be functional with as minimum as 2 MB of RAM and scale with as many gigabytes plugged into the machine.

Linux Cons:

1. Linux is not known to be as incredibly user-friendly as other commercial operating systems.

2. Since Linux is an open-source OS, there are far too many distributions; which might confuses new users about which distro to pick. You may think of a Linux Distribution as something similar to Windows Vista, Home, Professional, etc.

Why are we studying Linux?

For the short term, this module will be thought within a Linux context. Nevertheless, as a computer science graduate you are expected to deal with Linux at some point of your career; it is an operating system which is created by developers for developers.

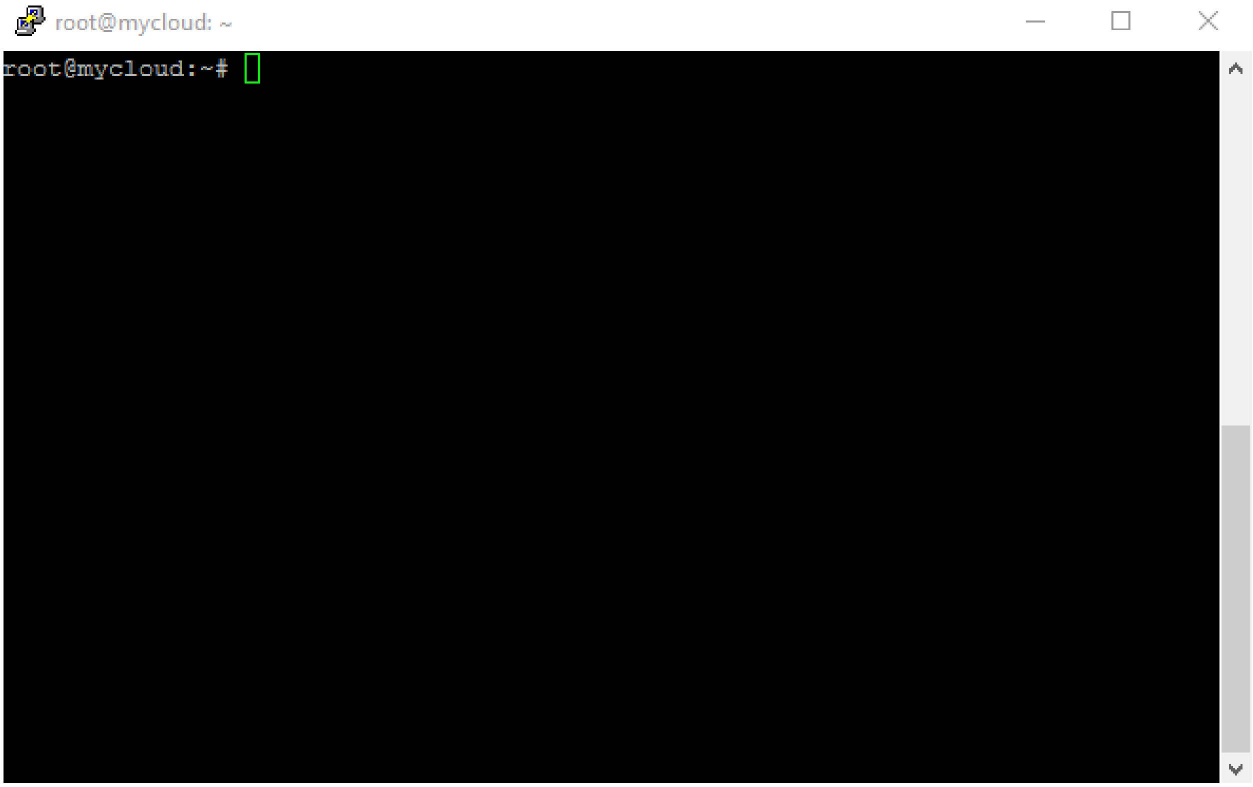
Are we expected to be Linux expertise after studying this module?

This module is essentially about understanding and applying different operating systems principles; for instance, how an OS actually handle all the tasks simultaneously (browsing, listening to music, playing video games, etc.) Linux is the context where we will be applying those concepts within, but it is not the core. Nevertheless, we will put you on the track of being a Linux expert whenever you need to; for this module we only require the basic knowledge of Linux.

So, now let’s dive into the Linux World!

The Shell

The shell acts as an interface between the user and the operating system (known as kernel while running). The shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt (% on our systems). In short, if you would like to create a folder (the folder term is defined as directory in Linux) you will have to write a command into the shell and execute it, similarly if you wish to delete, move, or even navigate to another directory.



```
root@mycloud:~#
```

Figure 1. Showing Linux Shell

Hands on Shell

As stated, we manipulate and navigate throughout the Linux operating system through executing set of commands. The list of full commands known as “Linux Cheat Sheet” – However, we only require knowing the most important ones.

Each command comes with options which can be used based on each use case, for instance, in Windows we can change the view of a folder content to “Detailed, Big Icons, Small Icons”

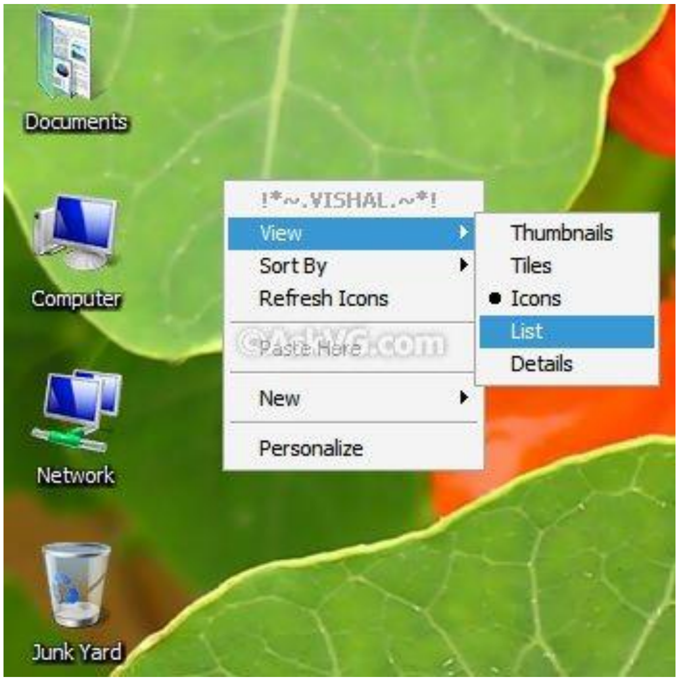


Figure 2. Showing Windows view options

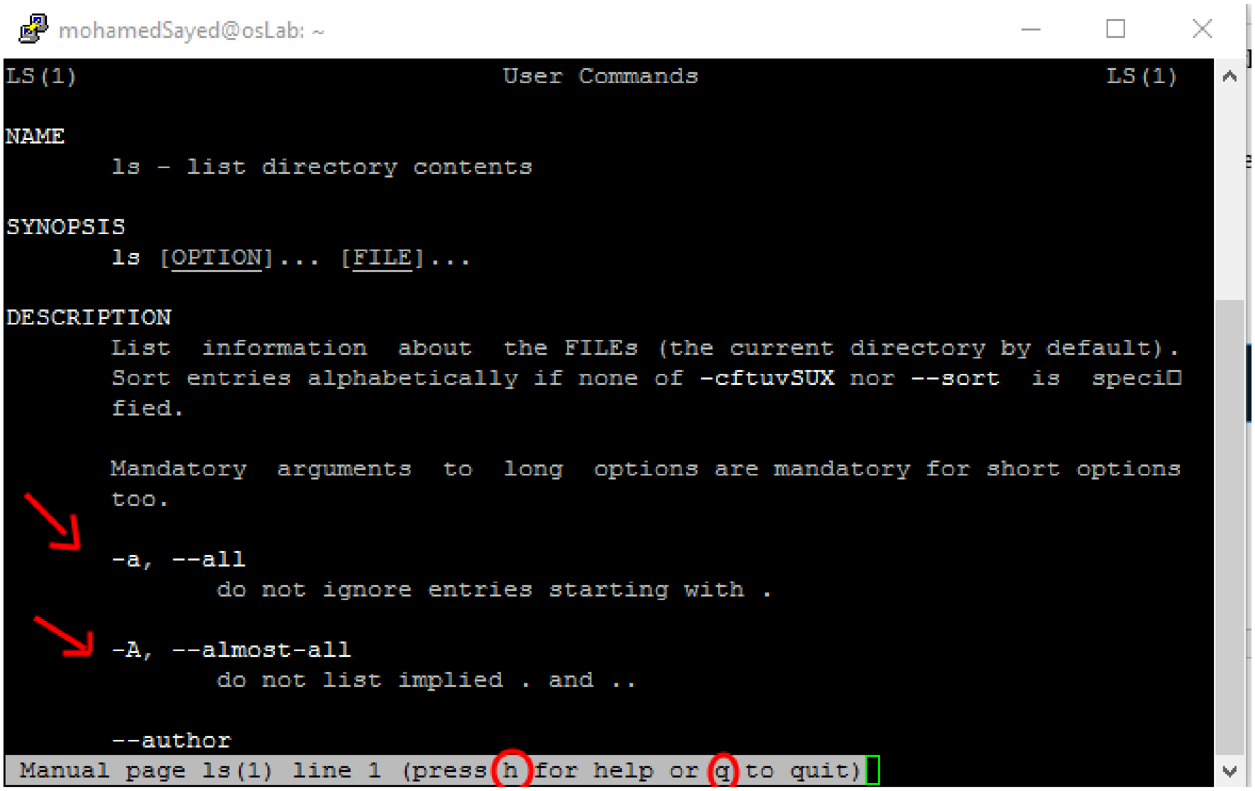
In Linux, we have similar approach through the arguments passed to each command.

Think of each command as a function which takes arguments (optionally) and based on each argument the output varies.

Q: How to know which arguments are available for each command and the effect each of them has on the output?

A: Each command comes with short documentation. Type `man` before the command name to view the documentation, for instance; `man ls`.

```
man ls
```



```
LS(1)                         User Commands                        LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci
       fied.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
 Manual page ls(1) line 1 (press h for help or q to quit)
```

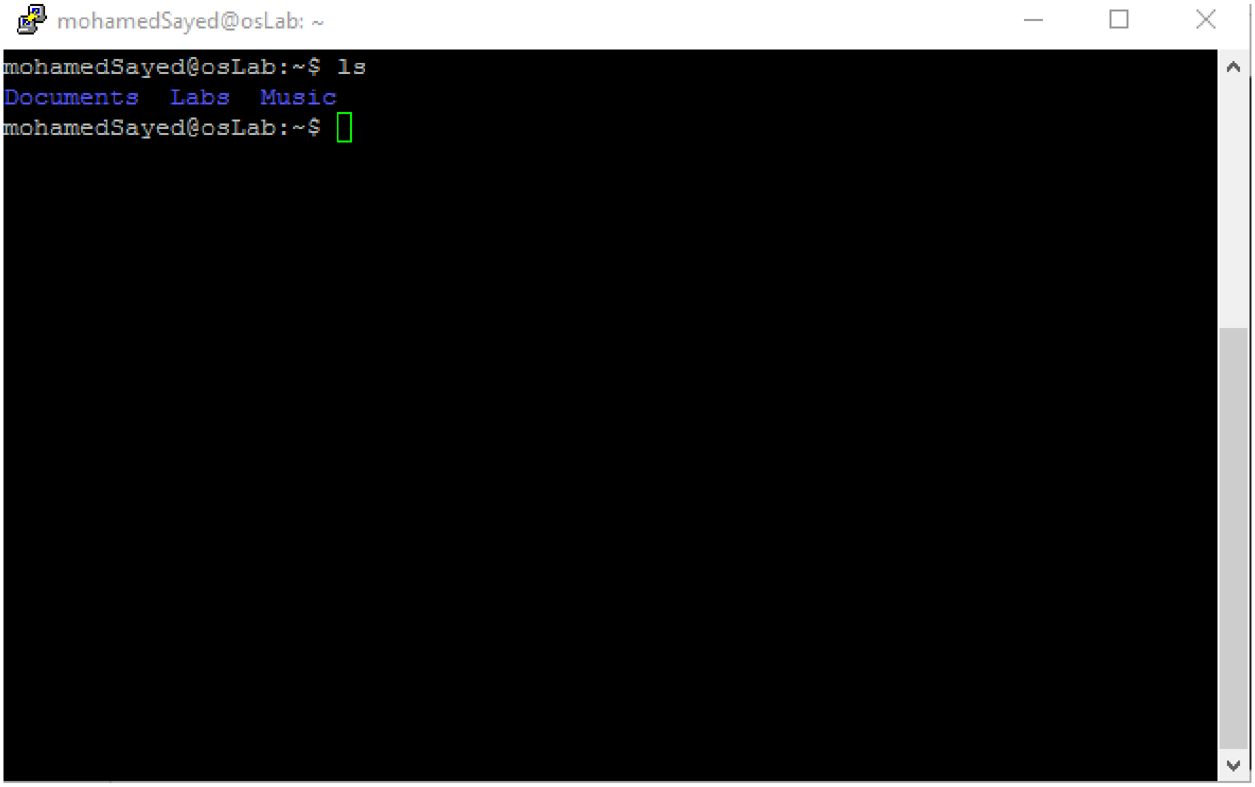
Figure 3. Showing man ls output.

Listing Files and Directories

When you first login, your current working directory is your home directory. Your home directory has the same name as your user-name, for example, **mohamedSayed**, and it is where your personal files and subdirectories are saved.

Type `ls` to view the content of your home directory.

```
ls
```



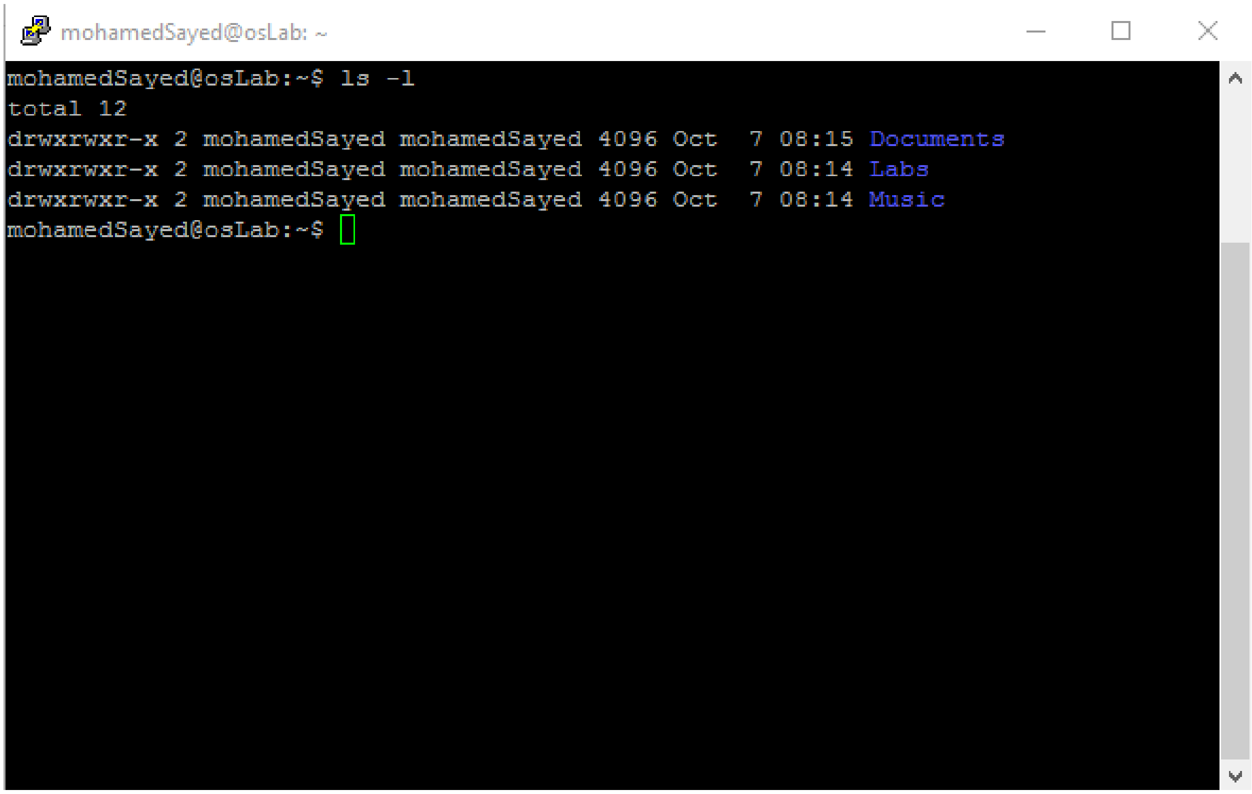
```
mohamedSayed@osLab:~$ ls
Documents  Labs  Music
mohamedSayed@osLab:~$
```

Figure 4. Showing ls command output.

Now assume we would like to view the content of a directory in more details and sorted in rows instead of columns (one directory shown per row).

Try to pass the argument `-l` to your `ls` command. The final command should be like this

```
ls -l
```



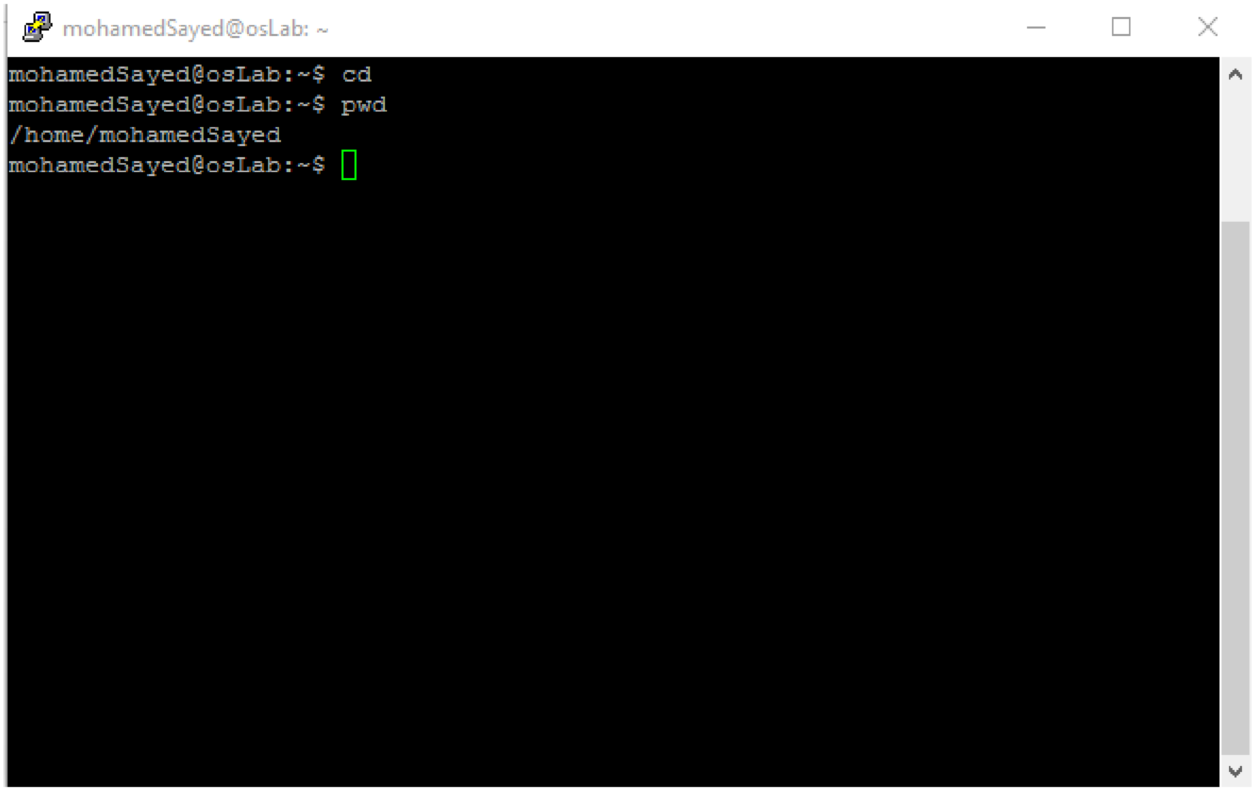
```
mohamedSayed@osLab:~$ ls -l
total 12
drwxrwxr-x 2 mohamedSayed mohamedSayed 4096 Oct  7 08:15 Documents
drwxrwxr-x 2 mohamedSayed mohamedSayed 4096 Oct  7 08:14 Labs
drwxrwxr-x 2 mohamedSayed mohamedSayed 4096 Oct  7 08:14 Music
mohamedSayed@osLab:~$
```

Figure 5. Showing ls -l command output.

Pathnames (Where Am I located right now!)

Pathnames enable you to work out where you are in relation to the whole file-system. For example, to find out the absolute pathname of your home-directory, type `cd` to get back to your home-directory and then type `pwd`

```
cd
```



```
mohamedSayed@osLab:~$ cd
mohamedSayed@osLab:~$ pwd
/home/mohamedSayed
mohamedSayed@osLab:~$
```

Figure 6. Showing pwd command output.

```
pwd
```

Exercise: Use the commands `cd`, `ls` and `pwd` to explore the file system.

(Remember, if you get lost, type cd by itself to return to your home-directory).

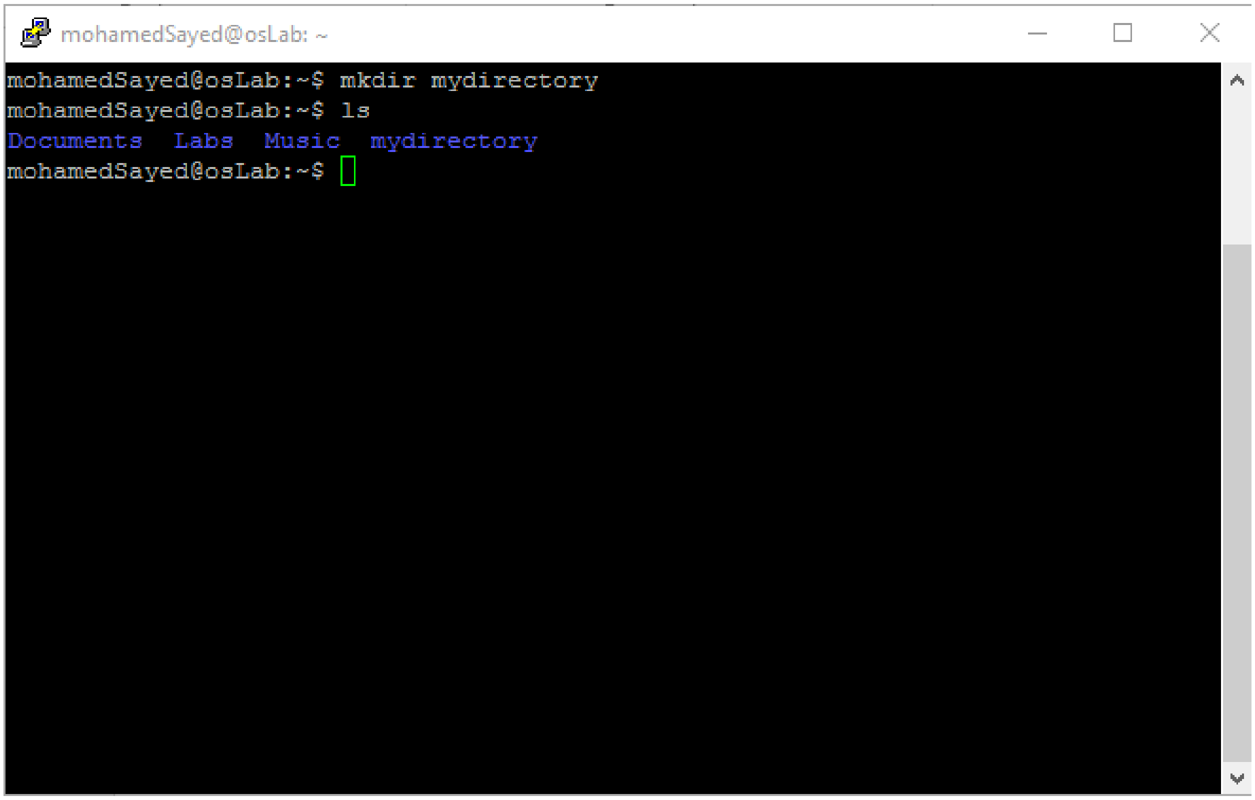
Creating Directories

mkdir (make directory) is the command used for directory creation.

```
mkdir mydirectory
```

Now type ls to confirm the directory creation.

```
ls
```



```
mohamedSayed@osLab:~$ mkdir mydirectory
mohamedSayed@osLab:~$ ls
Documents  Labs  Music  mydirectory
mohamedSayed@osLab:~$
```

Figure 7. Showing mdkir command output.

The success of your command execution is indicated by the terminal opening for executing further commands. In short, if no error messages have been displayed, it is an implicit saying of “Success”.

Navigating through directories

cd (change directory)

The command cd directory means change the current working directory to 'directory'. The current working directory may be thought of as the directory you are in, i.e. your current position in the file-system tree.

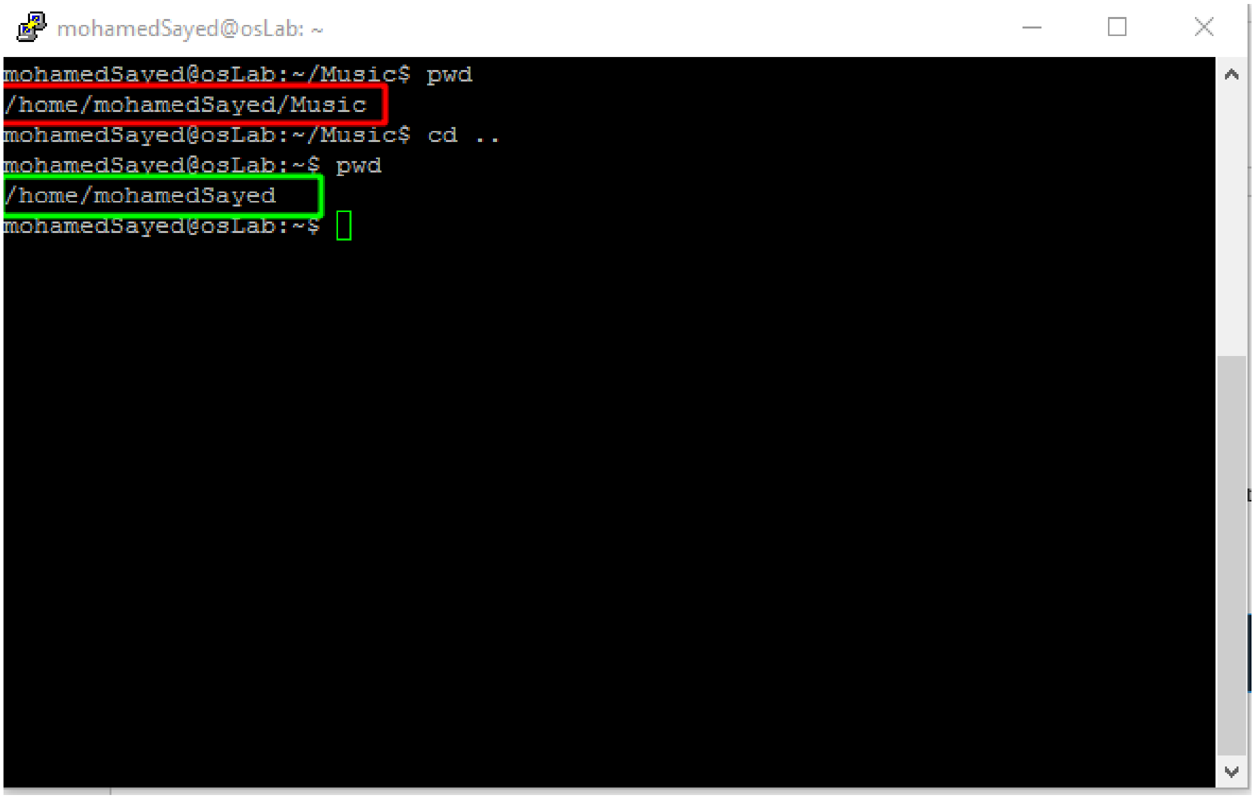
```
cd Music
```

Exercise: create a new directory using `mkdir` and navigate to it through `cd yourDirecotryName`

Change to parent directory (back forward)

After executing `cd  Music`, now your current working directory is set to `/home/mohamedSayed/Music` – you may reach the parent directory or previous one by executing `cd ..`

```
cd ..
```



mohamedSayed@osLab: ~

```
mohamedSayed@osLab:~/Music$ pwd
/home/mohamedSayed/Music
mohamedSayed@osLab:~/Music$ cd ..
mohamedSayed@osLab:~$ pwd
/home/mohamedSayed
mohamedSayed@osLab:~$
```

Figure 8. Demonstrating navigation from and to a parent directory.

Creating Files

In order to create a file we use the command `touch`

```
touch myfile
```

Note that myfile will be created at your current working directory.

Writing into Files

In order to write data into a file, use the `echo` command.

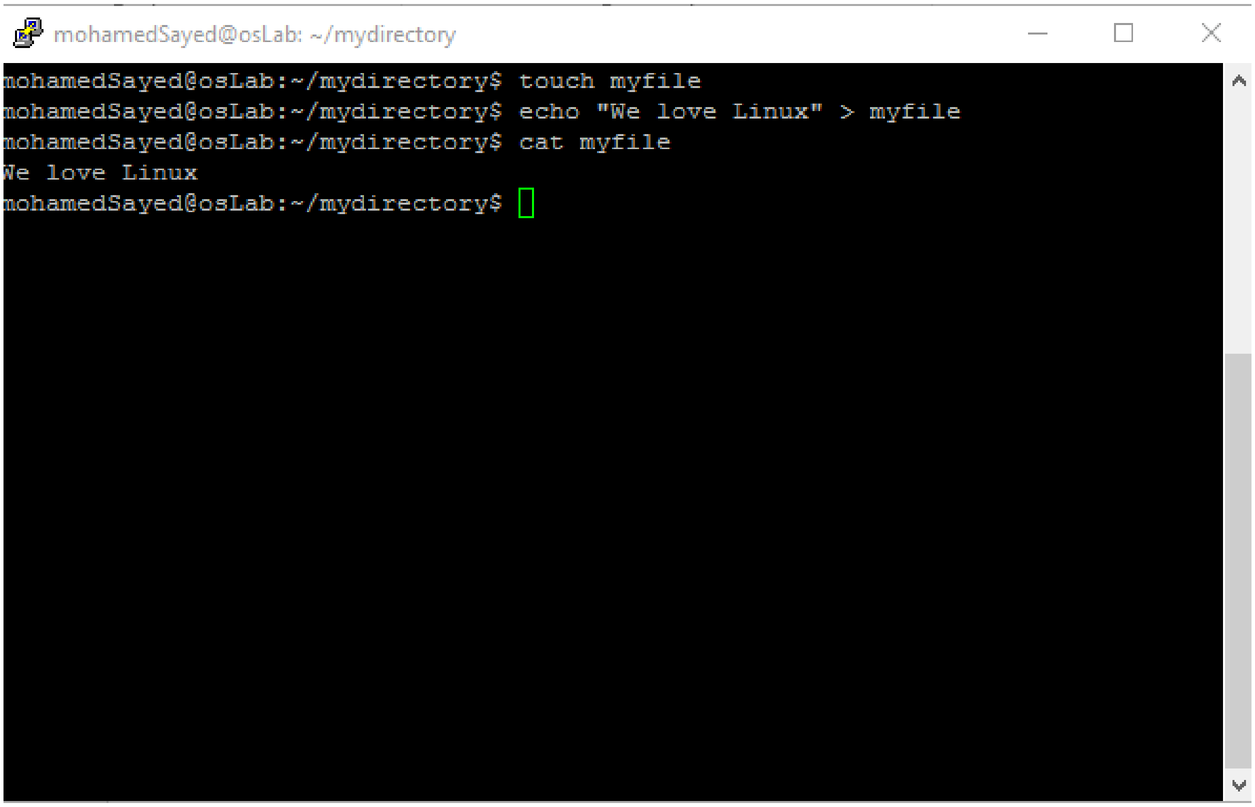
```
echo “Hello World” > myfile
```

Notice the “>” operator indicating that whatever text between the double quotes will be written into myfile. If the file did not exists, it will be automatically created.

Reading Files

In order to display the content of a file, we use the command `cat`

```
cat myfile
```



```
mohamedSayed@osLab:~/mydirectory$ touch myfile
mohamedSayed@osLab:~/mydirectory$ echo "We love Linux" > myfile
mohamedSayed@osLab:~/mydirectory$ cat myfile
We love Linux
mohamedSayed@osLab:~/mydirectory$
```

Figure 9. Demonstrating a combination of touch, echo, and cat.

Deleting Directories

In order to delete a directory we use the command `rmdir`

```
rmdir mydirectory
```

Deleting Files

In order to delete a file we use the command `rm`. rm removes each specified file. By default, it does not remove directories.

```
rm myfile
```

However, if the user passes the –r -R, or --recursive are given, then rm prompts the user for whether to proceed with the entire operation.

```
rm -r mydirectory
```

Clearing Shell

In order to clear screen from displayed commands use the `clear` command.

```
clear
```

Exercise One

1) Navigate to your absolute path by typing `cd`
2) Know your current working directory by executing `pwd`
3) Create new directory named as "test"
4) Confirm the creation of your created directory "test" by executing command `ls`.
5) Navigate to your created directory using `cd` command.
6) Confirm the success of your navigation operation by checking your current working directory using `pwd` command.
7) Create a file (not a directory) name as "myfile".
8) Write "Hello World" into "myfile".
9) Display the content of "myfile".

Exercise Two

1) Navigate to your absolute path by typing `cd`
2) Know your current working directory by executing `pwd`

3) Create new directory named as "test4" as child of test1/test2/test3. Hint: use the `man mkdir` and look after --parents

4) Navigate to "test4" directory within a **single cd operation**.

5) Navigate back from "test4" directory to "test" directory within a **single cd operation**.

6) Try to remove the directory "test" using `rmdir`. What is the output of the shell?

7) Use the man command to check the `rm` documentation for an argument to remove directories and their contents recursively, and then use that command to remove the "test" directory.

References

http://linoxide.com/guide/linux-command-shelf.html

http://linoxide.com/linux-command/linux-commands-cheat-sheet/

http://tldp.org/LDP/intro-linux/intro-linux.pdf

http://www.ee.surrey.ac.uk/Teaching/Unix/

https://www.techlila.com/introduction-linux-operating-system/