

# Assignment: Linux Basics Lab

## Section 1: Overview of Linux

A typical Linux operating system is made up of three parts: the kernel, the shell and the applications.

**Kernel:** The kernel of Linux is the hub of the operating system: it allocates time and memory to programs and handles the file storage as well as response to system calls.

**Shell:** The shell acts as an interface between the user and the kernel. When a user logs in (into command line only environment), the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI). It interprets the commands, which are typed by the user, and it arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt. A graphical user interface for the shell is called *Terminal*.

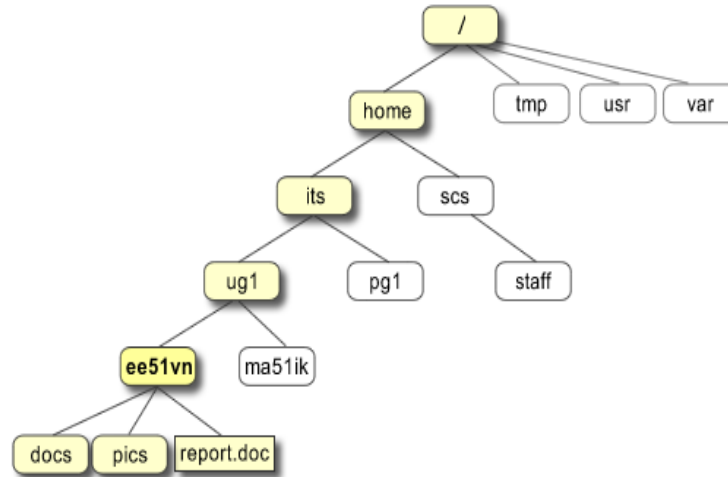
**Desktop Environment:** Linux operating systems also have a graphical user interface (GUI) through which users can access applications and system resources. Unlike other operating systems Linux OS has many Desktop Environments available. Users are free to install the GUI of their choice. Some popular ones are GNOME, KDE, XFCE, Ubuntu Unity, etc. When one installs a Linux system, it comes with a default GUI. One can install other GUIs at any time, but only one GUI environment can be used at a time. For example, Kali Linux comes with GNOME.

**Linux Directory Structure:** In Linux, files and directories (folders) are laid out on disk in hierarchical fashion in a tree structure. The top directory is called *root* folder, it is denoted by “/” (slash).

A collection of all files and directories in a Linux machine is referred to as a “filesystem”.

A path (location) of a file or directory in the filesystem is written starting from the root directory (/) followed by all intermediate directory names, which are separated by a slash with the actual file name (or directory name) in the end of the path. Example: the file path of the file “report.doc” in the above figure is written as “/home/its/ug1/ee51vn/report.doc”.

## Linux Basics and Capabilities Lab



### Notes:

1. In the path `/home/its/ug1/ee51vn/report.doc`, do not get confused between first slash `/` which denotes the root directory and subsequent slashes, which are used to separate directories at each level of the path.
2. There are no spaces in a file path. Example: The file path `/home/its/ug1/ee51vn/report.doc` is the one single string with no spaces between slashes and subsequent folder names.

### Q1: Write the path for directory *staff* in the above figure.

Different default directories under the `/` directory are used for some specific purposes.

**/bin** : All executable binary files (including commands) are present in this directory.

**/dev** : Contains device files for hardware devices (usb, cdrom etc.)

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

**/home** : Home directory of the users. Every time a new user is created, a directory with the corresponding username is created within `/home` which contains other directories such as *Desktop*, *Downloads*, *Documents*, etc.

**/lib** : Contains the *kernel modules* and *shared library* images required to boot the system and run commands in the root file system.

**/lost+found** : This directory is created during installation of Linux, useful for recovering files which may be broken due to unexpected shutdown.

**/media** : Temporary mount directory is created for removable devices viz., *media/cdrom*.

**/mnt** : Temporary mount directory for mounting file system.

**/opt** : Contains the third-party application software viz. Java, etc.

**/proc** : A virtual and pseudo filesystem which contains information about running processes with a particular process id abbreviated as "pid".

### Section 2: Introduction to Ubuntu Linux

There exist many Linux OS distributions (around 3500 are known), but most of them are created from the same Linux kernel. We will use *Ubuntu* in this course. Other popular Linux variants are Redhat, Debian, and CentOS.

#### Setup VirtualBox

If you already have setup virtualbox or any other linux environment , no need to follow this . But it should be an Ubuntu/Debian Based environment.

An OVA file is an Open Virtualization Appliance that contains a compressed, “installable” version of a virtual machine. When you open an OVA file, it extracts the VM and imports it into whatever virtualization software you have installed on your computer. The instructions below explain how to install an OVA file in Oracle VirtualBox. Briefly, these steps are:

1. If it is not already on your computer, download and install VirtualBox – <https://www.virtualbox.org/wiki/Downloads>. You can find instructions on how to do this at the Installing VirtualBox wiki page.
  - a. For Mac users, select “OS X hosts”. Run the file and follow the on-screen directions.
  - b. You can launch VirtualBox from the Applications and “Keep in dock” if desired.

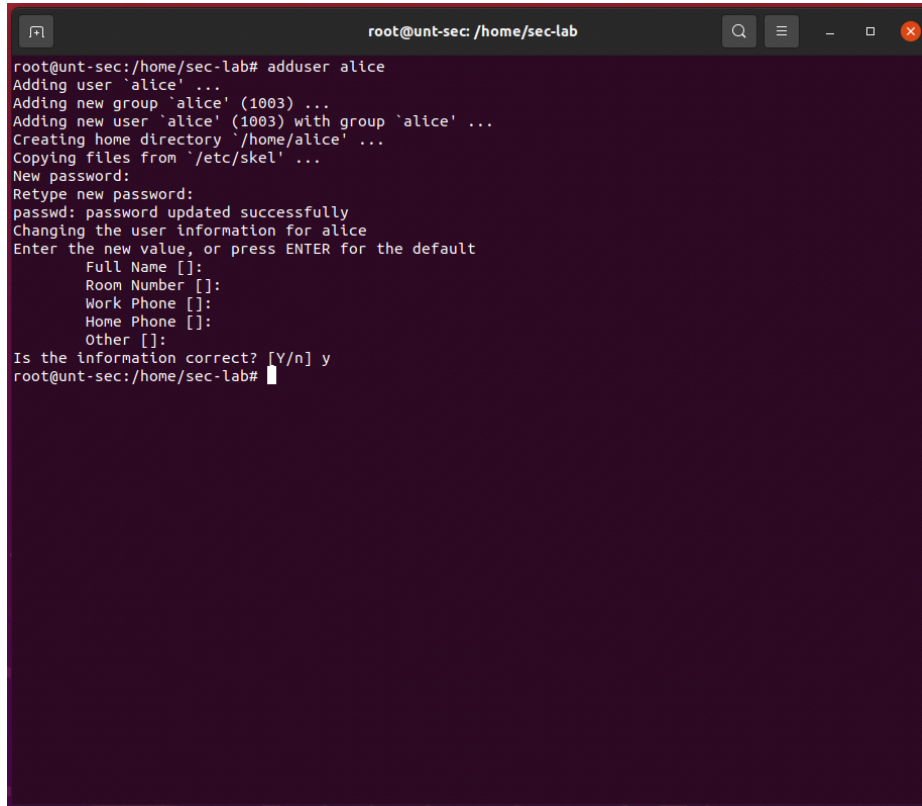
**IMPORTANT NOTE:** In the rest of this lab, you will replace “***EUID***” with below format <eg><YOUR\_REGISTRATION\_NUMBER>. Example: Let your registration number be “2894”; when instructed to type a command “**adduser *EUID***”, you will type “**adduser eg2894**”, and when instructed to type a command “**adduser *EUIDnew***”, you will type “**adduser eg2894new**”.

## User Account Management

1. Create a new user account (for yourself):
  - a. Type “sudo su”. The system will ask for the root password. Type “efacuor” as the password. The shell prompt will change to # and you are working now as a user root.
  - b. Type “**adduser *eid***”.
  - c. Enter a password for your account. You have to do it twice.

**Note:** Passwords are not shown in terminal as you type them. Usually while typing password you’ll see \*\*\*\*\* (stars) or in ..... (dots). But here you don’t see anything on a Linux Terminal.
  - d. Add further user details. These details are optional-you can just hit enter to skip for Full Name, Number, Phone etc.) –See the below screenshot.
  - e. Type “y” at “Is this information correct?”

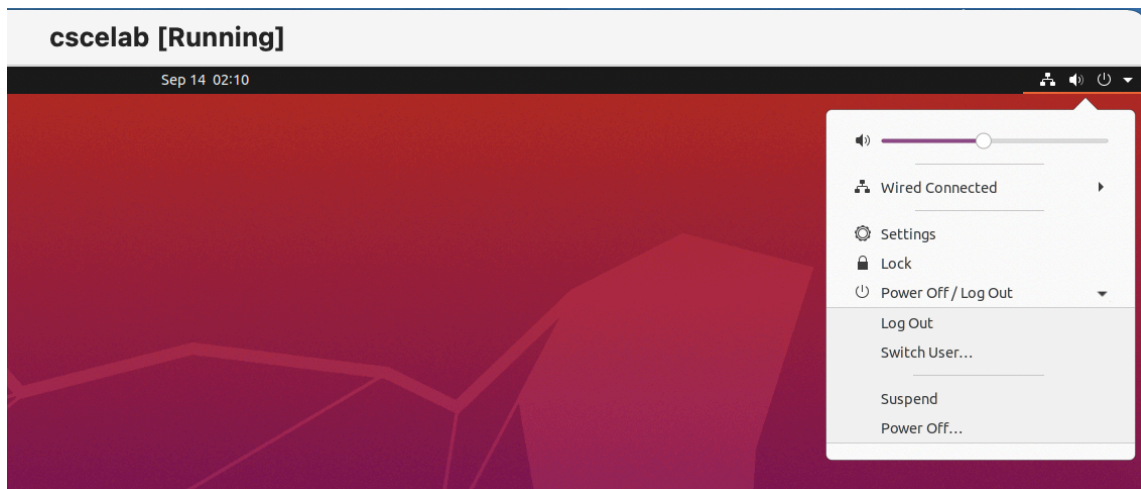
## Linux Basics and Capabilities Lab



```
root@unt-sec: /home/sec-lab
root@unt-sec:/home/sec-lab# adduser alice
Adding user 'alice' ...
Adding new group 'alice' (1003) ...
Adding new user 'alice' (1003) with group 'alice' ...
Creating home directory '/home/alice' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for alice
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@unt-sec:/home/sec-lab#
```

2. To change a user password, use **passwd** command.  
Example: **passwd *euid***
3. Add new user to sudoers file to give them superuser permissions.
  - a. Type “**nano /etc/sudoers**”. (Here, “nano” is a command-line text editor - details are provided in the Text Editors section of this manual.)
  - b. In this file, add the following line:  
*Username* ALL=(ALL:ALL) ALL. Add this line next to the following one:  
**root ALL=(ALL:ALL) ALL**  
You have to provide your newly created username.
  - c. Press Control+O, Press Enter, when asked for confirmation, Press Control+X to exit
4. Close the terminal window.
5. Go to the upper menu and expand the dropdown, expand the Power Off/ Log Out menu and then select Power Off...

## Linux Basics and Capabilities Lab



Then, in the dialog box, select the Restart button to restart.

Now select the newly created account and login. You may Skip and click on Next until you're able to select Done to complete the setup of the new account.

**Q2: Login to your new user account, open a terminal and type “sudo su”, enter sudo password for your account, and submit a screenshot. The screenshot should clearly show the prompt. It will look like “root@usudont-sec:/home/euid”.**

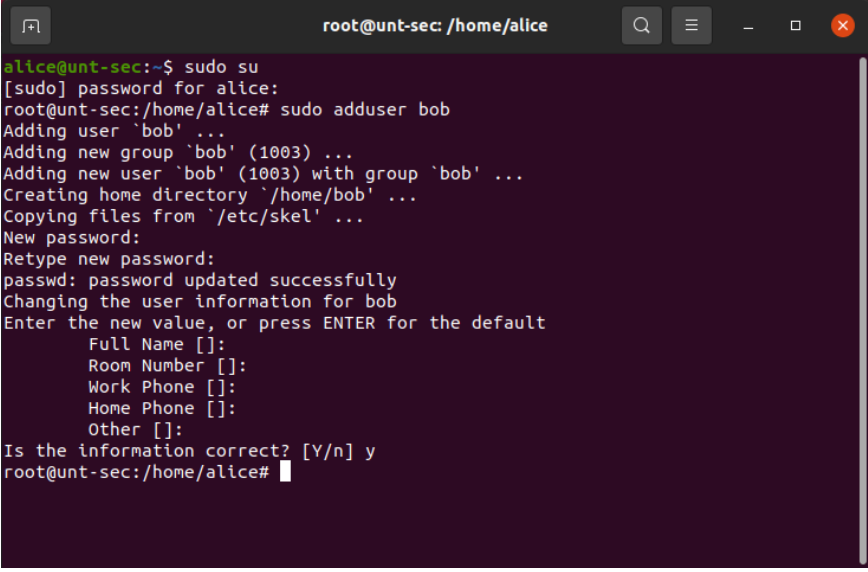
6. Create another user from your account.

- Open Terminal after logging into your account.
- Type “**sudo adduser *euidnew***”. Enter sudo password for your account.

**Note:** the user password is asked when you use the “sudo” command for first time in a terminal. “sudo” is used to elevate a normal user to root user for execution certain high privilege commands.

- Add a (unix) password for new user ***euidnew*** twice and rest of the details as earlier.

## Linux Basics and Capabilities Lab



```
root@unt-sec: /home/alice
alice@unt-sec:~$ sudo su
[sudo] password for alice:
root@unt-sec:/home/alice# sudo adduser bob
Adding user 'bob' ...
Adding new group 'bob' (1003) ...
Adding new user 'bob' (1003) with group 'bob' ...
Creating home directory '/home/bob' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for bob
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@unt-sec:/home/alice#
```

7. To change to root user from normal user (type “**exit**” if currently in root mode for user):
  - a. Type “**sudo su -**”.
  - b. If prompted, type the root user password at the prompt. After successful login, you should observe a change in shell prompt from your user to root user.
  - c. Type “**exit**” to exit from the root user. Now you should be back to you user prompt.
8. Type “tail /etc/passwd” (note that “/etc/passwd” is one string—no spaces in between).

### Q3: Attach a screenshot of the result.

A *root* user (also referred to as a *superuser*) is an administrator of the system who has full privileges by default, while normal users have limited privileges. Normal users need to use the “sudo” command in order to temporarily gain certain high-privilege commands. Example of these commands are: software installation, changing to the root user in a terminal, changing the system settings, and others.

### Some Basic Commands

#### Files and Directories

##### 9. pwd – present working directory

Type “**pwd**” at command prompt (without quotes). This command displays the directory in which you are currently. When a user starts a shell the default directory is user’s home directory: `/home/euid`.

##### 10. cd – change directory

This command is used to change to a desired directory from current directory.

- a. Type “**cd /home**” (without quotes). This will move the user to the `/home` directory.
- b. Type “**pwd**”. What is it showing
- c. Type “**cd -**”. This will take you back to previous directory, i.e., your home directory
- d. You can use “**cd ~**” to go back to your home directory from. (`~` Tilde symbol)
- e. `.` (single dot) refers to current directory, `..` (double dots) refer to parent directory.  
“**cd .**” keeps you in same directory, while “**cd ..**” moves to parent directory, a level above. “**cd ../../**” moves you two levels above the current directory.

##### 11. ls – list

**ls** command simply lists file and directories in present directories. Option “**-l**” which gives more details about files which include file size, time created, file permissions etc

- a. Type “**ls -l**”  
To display details of a directory or file: `ls -l directory or file name`. Ex: `ls -l Desktop`
- b. Type “**ls -la**”

**Q4: What output “ls -la” gives (take a screenshot)?**

**Q5: What difference did you observe between the results of “ls -l” and “ls -la”?  
What does the option “-a” mean?**

**Hint:** Refer to the *man* page: `man ls`. Also, see the section on *man help* below.

##### 12. mkdir - create a directory

- a. Type “**mkdir firstdir**”. It creates a new directory called `firstdir`. You can also create multiple directories and also nested directories
- b. Type “**mkdir dir1 dir2 dir3**.” It creates 3 directories in present directory.
- c. Type “**mkdir -p dir11/dir22/{dir33, dir44}**”.  
It creates 4 directories: `dir11`, `dir22` inside `dir11`, `dir33` and `dir44` inside `dir22`.
- d. Type “**ls**”



## Linux Basics and Capabilities Lab

### 13. rm - delete a file or directory

- type “**rm -r dir3**”. You can also use “**rmdir**” but it only deletes empty directories

### 14. touch - create empty file(s)

- Type “**touch file1 file2 file3**”.

### 15. mv - move or rename files and directories

- a. Type “**mv file1 dir1**” - this moves (cut and paste) the file *file1* to the directory *dir1*.
- b. Type “**mv file2 file22**”.
- c. Type “**ls**”.

### 16. cp - copy files and directories

Type the following commands in the order shown below:

- “**touch file4**”. Creates a file with the name *file4*.
- “**mkdir dir4**”. Creates a directory *dir4*.
- “**cp file4 dir4**”. Copies *file4* to *dir4*.
- “**mkdir dir5**”. Creates another directory *dir5*.
- “**cp -r dir4 dir5**”. Copies *dir4* to *dir5*.

**Q6: Why do we have to use “-r” option to copy directories? (Hint: Use man cp.)**

### 17. find - to find files and directories.

- Type “**find /root -name file1**”.

**Q7: What is the result? (Show the screenshot.)**

You are searching for the file with name *file1* in the directory */root*. This command will search for the file in */root* and in all its subdirectories. You gave */root* as search location as you think *file1* must be somewhere in it. If you have no information where the file might be in your system, you should start searching for it from the most top directory in the filesystem, that is “/”. In this later case, the command should be **find / -name file1**.

18. The absolute path of a file or directory is its path written from the root of the file system. For example, consider the directory *dir44* you created earlier in Step 5. Its absolute path is */home/euid/dir11/dir22/dir44*.
19. The relative path of a file or directory is its path from the current directory. Now, you are in your home directory which is */home/euid*. The relative path of *dir44* from this directory is *dir11/dir22/dir44*. One writes the relative path starting from a directory one level below, i.e., *dir11* in our case.

## Linux Basics and Capabilities Lab

20. One can refer to a file or directory either using relative or absolute path (both are possible).

- Type **“touch /home/euid/dir11/dir22/dir44/file44”**.  
(Replace *username* you actual username).
- Type **“dir11dir22/dir44/file55”**

Either way you are able can access dir44

**Note:** When using the relative path, one should be aware of the current directory. In the above example, when the current directory changes to *home*, then the relative path to dir44 changes to *euid/dir11/dir22/dir44*.

### File Compression and Archiving

21. To compress and archive files: In Linux file compression and archiving (grouping file together) and compression are two different activities. First, we archive multiple files with “tar” command. A file with “.tar” extension is created. Then we zip (compress the file) with gzip command. (gzip-stands for gunzip is compression tool in linux. Bunzip (bzip2) and zip are other commonly used compression tools)

- a. Type **“touch a b c d”**. Creates 4 files *a*, *b*, *c* and *d*.
- b. **tar –cf compressed.tar a b c d**. We are creating a tar file **compressed.tar** from the files *a*, *b*, *c* and *d*.
- c. **gzip –vf compressed.tar**. This should create a file *compressed.tar.gz*.

**Note:** For “tar” command, you should supply output filename along with .tar extension. For gzip, it automatically creates a zipped file with .gz extension. You no longer have .tar file after you compressed it with gzip.

- d. You can also do it in single step which is usually the preferred way:  
type **“tar –zcvf compressed2.tar.gz a b c d”**.

22. To extract and uncompress files:

- a. Type **“rm –rf a b c d”**. Deleting the files.
- b. Type **“ls”**.
- c. Type **“gzip –dvf compressed.tar.gz”** – you will get the file *compressed.tar*.
- d. Type **“tar –xvf compressed.tar”**. You will get the original files *a*, *b*, *c* and *d*.
- e. Type **“ls”**.
- f. As before, you can uncompress and extract the *compressed2.tar.gz* in a sigle step:  
Type **“tar –zxvf compressed2.tar.gz”**

### Getting help with the man command

The “**man**” command is used to get help for any command in Linux. **man** stands for manual. To pull up a man page of a command, type **man command**. Ex: **man ls**. This command displays list of options available with the command. Man pages explain different options and syntax of a command and are usually difficult to understand for beginners. If you want to know usage and example for a command, google is the best place to look.

Type “**q**” to quit from man page.

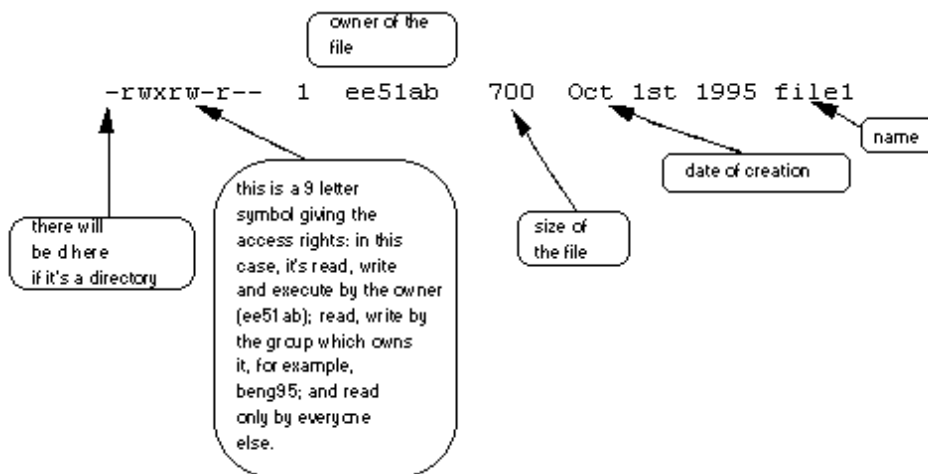
### Section 3: File Permissions and Access Control

In Linux a file has read, write and execute permissions. And permissions are assigned to users of three categories: **owner**, **group** and **others**. The user who creates a file will be the **owner**, **group** is group of users who has access. **Others** are all the users other than **owner** and users in the **group**.

#### 1. Understanding file/directory permissions:

Type `ls -l filename` to list the file permissions.

The below figure explains the file details displayed.



Each file (and directory) has associated access rights, which may be found by typing `ls -l`. Also, `ls -lg` gives additional information as to which group owns the file (beng95 in the following example):

```
-rwxrw-r-- 1 ee51ab beng95 2450 Sept29 11:52 file1
```

In the left-hand column is a 10 symbol string consisting of the symbols `d`, `r`, `w`, `x`, `-`, and, occasionally, `s` or `S`. If `d` is present, it will be at the left hand end of the string, and indicates a directory: otherwise `-` will be the starting symbol of the string.

## Linux Basics and Capabilities Lab

The 9 remaining symbols indicate the permissions, or access rights, and are taken as 3 groups of 3.

- The left group of 3 gives the file permissions for the user that owns the file (or directory) (ee51ab in the above example);
- the middle group gives the permissions for the group of people to whom the file (or directory) belongs (eebeng95 in the above example); For every user in Linux, a group with the same name is also created. This group will be default group on newly created files.
- the rightmost group gives the permissions for all others.

The symbols r, w, etc., have slightly different meanings depending on whether they refer to a simple file or to a directory.

### 2. Access rights on files.

- r (or -), indicates read permission (or otherwise), that is, the presence or absence of permission to read and copy the file
- w (or -), indicates write permission (or otherwise), that is, the permission (or otherwise) to change a file
- x (or -), indicates execution permission (or otherwise), that is, the permission to execute a file, where appropriate

### 3. Access rights on directories.

- **r** allows users to list files in the directory;
- **w** means that users may delete files from the directory or move files into it;
- **x** means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files.
- So, in order to read a file, you must have executed permission on the directory containing that file, and hence on any directory containing that directory as a subdirectory, and so on, up the tree.

#### Some Examples:

-rwxrwxrwx	a file that everyone can read, write and execute (and delete).
-rw-----	a file that only the owner can read and write - no-one else can read or write and no-one has execution rights (e.g., your mailbox file).

### 4. Changing access rights (i.e., permissions) on a file

#### a. **chmod** (changing a file mode):

- Only the owner of a file can use **chmod** to change the permissions of a file. The options of **chmod** are as follows

Symbol	Meaning
--------	---------

## Linux Basics and Capabilities Lab

u	user
g	group
o	other
a	all
r	read
w	write (and delete)
x	execute (and access directory)
+	add permission
-	take away permission

- ii. type “**touch access**” (to create a file called *access*).
- iii. type “**ls -l access**”.
- iv. To remove read write and execute permissions on the file **access** for the group and others type “**sudo chmod go-rwx access**”. This will leave the other permissions unaffected.
- v. To give read and write permissions on the file **access** to all, type “**sudo chmod a+rw access**”.
- b. **chgrp**-this command is used to change the group of a file
  - i. Type “**sudo chgrp *eidnew* access**”. This will change the default group on file “access” to *eidnew* from your current *username*.
  - ii. Type “**ls -l access**”.
- c. **chown**- this command is used change ownership of a file or directory. This can also be used to change the group like **chgrp**.
  - i. Type “**touch own\_file**”.
  - ii. Type “**ls -l own\_file**”.
  - iii. Type “**sudo chown *eidnew* own\_file**”. This change owner of the file to *eidnew* from you.
  - iv. Type “**mkdir own\_dir**”.
  - v. Type “**ls -ld own\_dir**”. The **-ld** option displays properties of directory. If you only use **-l** it shows properties of contents of **own\_dir**.
  - vi. Type “**sudo chown *eidnew:eidnew* own\_dir**”. This changes both the owner and the group to *eidnew* on folder **own\_dir**.
  - vii. Type “**ls -ld own\_dir**”.

### Q8: Submit a screenshot of the terminal.

**Note:** The command **chown** only changes the owner of a directory but not its contents. To change ownership of files and directories within a directory *own\_dir*, along with is

## Linux Basics and Capabilities Lab

permissions use **chown** with “-R” option.

Example: “**sudo chown -R *euclidnew: euclidnew* own\_dir**”.

### Section 4: Installing Software

1. To install a software package **apt-get** command is used in Ubuntu Linux.

- Type “**sudo apt-get update**”
- Type “**sudo apt-get install chromium**”  
(if this does not work, use “**sudo apt-get install chromium-browser**”.)

It checks for file size to be downloaded and ask for confirmation. Type “y” and hit enter.

This command installs Chromium web browser.

**Q9: Take a screenshot of the notification that installation is complete.**

**Note:** If you do not know the exact name of the package, you can search for it with “apt-cache” command. Example: **apt-cache search openoffice**. Relevant packages will be shown.

2. Installing software from GUI: One can also install software using “Add or Remove” utility.

Click **Applications → System Tools → Add or Remove Software**.

3. To download a file from the Internet, use **wget** command.

Type

**wget https://kekeeseen.files.wordpress.com/2013/03/linux\_commands\_08.pdf**

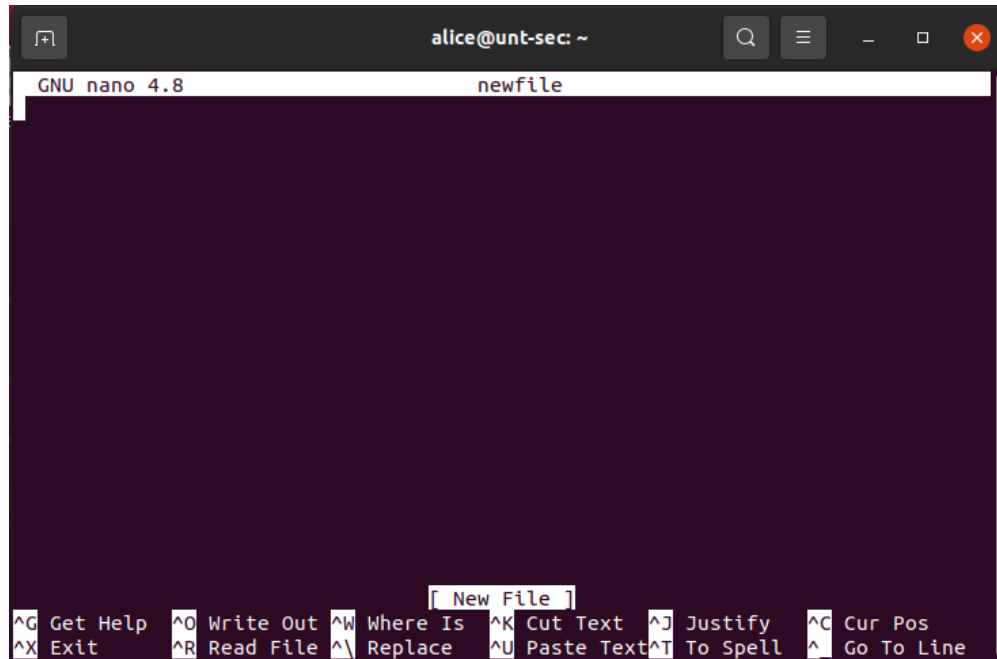
This will download the pdf file. Note that the wget utility works any file type.

### Section 5: Text Editors (Nano and Vim)

#### Nano:

- a. **To open a file in nano:** Type “**nano newfile**”. This will open a blank file called newfile.  
If there is already a file by name **newfile** nano will open that. Only if there exists no file by the name supplied it opens a new blank file with the supplied name.
- b. **To edit and save:** To write data to file, move cursor with arrow keys to location where you want to add content and start writing. Delete content using backspace.  
Press **Ctrl+O**, it ask for confirmation: **File name to write: newfile**. Hit Enter.
- c. **To exit nano:** Press **Ctrl+X**.
- d. **To search for a string or keyword:** Press **Ctrl+W**. A **search:** prompt appears at the bottom, type keyword you want to search and press enter.  
**Note:** You have all the above explained controls at the bottom of the **nano** editor with brief description.

## Linux Basics and Capabilities Lab



### Vim:

**VIM Editor** has 2 different modes: Command Mode and Insert Mode.

- a. **Command mode:** Type: **"vim blankfile"**. This opens a new file-blank file if a file doesn't exist by the name. A file is opened into command mode by default. In this mode you can issue commands to edit text.
- b. **Insert mode:**  
Hit letter **"i"**, which moves to insert mode from command mode. You can edit, add content to the file in this mode. Use arrow keys to navigate, backspace to delete.  
To save changes to the file:
  - i. Press Escape key-it takes file back to command mode.
  - ii. Press **"Shift + :"** A : prompt is created at bottom of file.
  - iii. Type letter **"w"**-it saves your file  
**Note:** If you don't want save the file at this point but want to go back, hit escape-to go to command mode and then **"i"** to go to insert mode. You always have to be in command mode before you go to insert mode.
  - iv. Again Press **"Shift + :"**- and type **"q"** to exit vi editor. If file has unsaved modifications, **vim** asks for confirmation. Typing **q!** Forcefully quits without saving.
- c. **More about command mode:** This mode takes commands from key board, to do operation like copy, paste, cut delete, etc. You can navigate using arrow keys.
  - i. **dd** - delete current line (in which cursor is present). **ndd** -deletes n number of lines. Example: **5dd**.

## Linux Basics and Capabilities Lab

- ii. **yy** - copy current line (in which cursor is present). **nyy**- copies n number of lines.
- iii. **p** - paste copied line(s) below current line.

**d.** To print content of text file to terminal

- i. Type “**cat newfile**”. It will dump all the content in *newfile* to terminal.
- ii. Type “**less newfile**”. This will open newfile on terminal but only partially display its content. Press enter to browse the file one line at a time. Press **Q** to exit less.

There exist many other text editors such as **emacs**, **pico**, and others, however we limit our demonstration to **nano** and **vim**. The main reasons are that they are simple, and that they are likely to be present (built-in) in various current and future Linux/Unix distributions. Typically, these operating systems have GUI-based text editors available as well.

**Your Final Task :**

1. Install AWS CLI on your linux environment
2. Configure AWS Credentials in your linux environment
3. Write a python script using boto3 to list the EC2 instances , your script should print each instance with it's status. If they are running , your script should stop them.
4. Include relevant screenshots here with the descriptions



### Appendix: List of Useful Commands

**Note:** The below list is not exhaustive: it is composed to cover some of the commands, which will be commonly used in our labs. The students are recommended to use the command “man” in order to see a complete list of available options.

a

alias Create an alias of a command  
apropos Search Help manual pages (man -k)  
apt-get Search for and install software packages (Debian/Ubuntu for Centos use yum)  
aptitude Search for and install software packages (Debian/Ubuntu)  
aspell Spell Checker  
awk Find and Replace text, database sort/validate/index

b

basename Strip directory and suffix from filenames  
bash GNU Bourne-Again SHell  
bc Arbitrary precision calculator language  
bg Send to background  
break Exit from a loop •  
builtin Run a shell builtin  
bzip2 Compress or decompress named file(s)

c

cal Display a calendar  
case Conditionally perform a command  
cat Concatenate and print (display) the content of files  
cd Change Directory  
cfdisk Partition table manipulator for Linux  
chgrp Change group ownership  
chmod Change access permissions  
chown Change file owner and group  
chroot Run a command with a different root directory  
chkconfig System services (runlevel)  
cksum Print CRC checksum and byte counts  
clear Clear terminal screen  
cmp Compare two files  
comm Compare two sorted files line by line  
command Run a command - ignoring shell functions •  
continue Resume the next iteration of a loop •  
cp Copy one or more files to another location  
cron Daemon to execute scheduled commands  
crontab Schedule a command to run at a later time  
csplit Split a file into context-determined pieces  
cut Divide a file into several parts

d

## Linux Basics and Capabilities Lab

date Display or change the date & time  
dc Desk Calculator  
dd Convert and copy a file, write disk headers, boot records  
ddrescue Data recovery tool  
declare Declare variables and give them attributes •  
df Display free disk space  
diff Display the differences between two files  
diff3 Show differences among three files  
dig DNS lookup  
dir Briefly list directory contents  
dircolors Colour setup for `ls`  
dirname Convert a full pathname to just a path  
dirs Display list of remembered directories  
dmesg Print kernel & driver messages  
du Estimate file space usage

e

echo Display message on screen •  
egrep Search file(s) for lines that match an extended expression  
eject Eject removable media  
enable Enable and disable builtin shell commands •  
env Environment variables  
ethtool Ethernet card settings  
eval Evaluate several commands/arguments  
exec Execute a command  
exit Exit the shell  
expect Automate arbitrary applications accessed over a terminal  
expand Convert tabs to spaces  
export Set an environment variable  
expr Evaluate expressions

f

false Do nothing, unsuccessfully  
fdformat Low-level format a floppy disk  
fdisk Partition table manipulator for Linux  
fg Send job to foreground  
fgrep Search file(s) for lines that match a fixed string  
file Determine file type  
find Search for files that meet a desired criteria  
fmt Reformat paragraph text  
fold Wrap text to fit a specified width.  
for Expand *words*, and execute *commands*  
format Format disks or tapes  
free Display memory usage  
fsck File system consistency check and repair  
ftp File Transfer Protocol

## Linux Basics and Capabilities Lab

function Define Function Macros

fuser Identify/kill the process that is accessing a file

g

gawk Find and Replace text within file(s)

getopts Parse positional parameters

grep Search file(s) for lines that match a given pattern

groupadd Add a user security group

groupdel Delete a group

groupmod Modify a group

groups Print group names a user is in

gzip Compress or decompress named file(s)

h

hash Remember the full pathname of a name argument

head Output the first part of file(s)

help Display help for a built-in command •

history Command History

hostname Print or set system name

i

iconv Convert the character set of a file

id Print user and group id's

if Conditionally perform a command

ifconfig Configure a network interface

ifdown Stop a network interface

ifup Start a network interface up

import Capture an X server screen and save the image to file

install Copy files and set attributes

j

jobs List active jobs •

join Join lines on a common field

k

kill Stop a process from running

killall Kill processes by name

l

less Display output one screen at a time

let Perform arithmetic on shell variables •

link Create a link to a file

ln Create a symbolic link to a file

local Create variables •

locate Find files

logname Print current login name

logout Exit a login shell •

look Display lines beginning with a given string

lpc Line printer control program

lpr Off line print

## Linux Basics and Capabilities Lab

lprint Print a file  
lprintd Abort a print job  
lprintq List the print queue  
lprm Remove jobs from the print queue  
ls List information about file(s)  
lsuf List open files

### m

make Recompile a group of programs  
man Help manual  
mkdir Create new folder(s)  
mkfifo Make FIFOs (named pipes)  
mkisofs Create an hybrid ISO9660/JOLIET/HFS filesystem  
mknod Make block or character special files  
more Display output one screen at a time  
mount Mount a file system  
mtools Manipulate MS-DOS files  
mtr Network diagnostics (traceroute/ping)  
mv Move or rename files or directories  
mmv Mass Move and rename (files)

### n

netstat Networking information  
nice Set the priority of a command or job  
nl Number lines and write files  
nohup Run a command immune to hangups  
notify-send Send desktop notifications  
nslookup Query Internet name servers interactively

### o

open Open a file in its default application  
op Operator access

### p

passwd Modify a user password  
paste Merge lines of files  
pathchk Check file name portability  
ping Test a network connection  
pkill Stop processes from running  
popd Restore the previous value of the current directory  
pr Prepare files for printing  
printcap Printer capability database  
printenv Print environment variables  
printf Format and print data •  
ps Process status  
pushd Save and then change the current directory  
pv Monitor the progress of data through a pipe  
pwd Print Working Directory

## Linux Basics and Capabilities Lab

q

quota Display disk usage and limits  
quotacheck Scan a file system for disk usage  
quotactl Set disk quotas

r

ram ram disk device  
rcp Copy files between two machines  
read Read a line from standard input •  
readarray Read from stdin into an array variable •  
readonly Mark variables/functions as readonly  
reboot Reboot the system  
rename Rename files  
renice Alter priority of running processes  
remsync Synchronize remote files via email  
return Exit a shell function  
rev Reverse lines of a file  
rm Remove files  
rmdir Remove folder(s)  
rsync Remote file copy (Synchronize file trees)

s

screen Multiplex terminal, run remote shells via ssh  
scp Secure copy (remote file copy)  
sdiff Merge two files interactively  
sed Stream Editor  
select Accept keyboard input  
seq Print numeric sequences  
set Manipulate shell variables and functions  
sftp Secure File Transfer Program  
shift Shift positional parameters  
shopt Shell Options  
shutdown Shutdown or restart linux  
sleep Delay for a specified time  
slocate Find files  
sort Sort text files  
source Run commands from a file '.'  
split Split a file into fixed-size pieces  
ssh Secure Shell client (remote login program)  
strace Trace system calls and signals  
su Substitute user identity  
sudo Execute a command as another user  
sum Print a checksum for a file  
suspend Suspend execution of this shell •  
sync Synchronize data on disk with memory

t

## Linux Basics and Capabilities Lab

tail Output the last part of file  
tar Store, list or extract files in an archive  
tee Redirect output to multiple files  
test Evaluate a conditional expression  
time Measure Program running time  
timeout Run a command with a time limit  
times User and system times  
touch Change file timestamps  
top List processes running on the system  
traceroute Trace Route to Host  
trap Run a command when a signal is set(bourne)  
tr Translate, squeeze, and/or delete characters  
true Do nothing, successfully  
tsort Topological sort  
tty Print filename of terminal on stdin  
type Describe a command •

u

ulimit Limit user resources •  
umask Users file creation mask  
umount Unmount a device  
unalias Remove an alias •  
uname Print system information  
unexpand Convert spaces to tabs  
uniq Uniquify files  
units Convert units from one scale to another  
unset Remove variable or function names  
unshar Unpack shell archive scripts  
until Execute commands (until error)  
uptime Show uptime  
useradd Create new user account  
userdel Delete a user account  
usermod Modify user account  
users List users currently logged in  
uuencode Encode a binary file  
uudecode Decode a file created by uuencode

v

v Verbosely list directory contents ('ls -l -b')  
vdir Verbosely list directory contents ('ls -l -b')  
vi Text Editor  
vmstat Report virtual memory statistics

w

wait Wait for a process to complete •  
watch Execute/display a program periodically  
wc Print byte, word, and line counts

## Linux Basics and Capabilities Lab

whereis Search the user's \$path, man pages and source files for a program

which Search the user's \$path for a program file

while Execute commands

who Print all usernames currently logged in

whoami Print the current user id and name ('id -un')

wget Retrieve web pages or files via HTTP, HTTPS or FTP

write Send a message to another user

x

xargs Execute utility, passing constructed argument list(s)

xdg-open Open a file or URL in the user's preferred application.

yes Print a string until interrupted

zip Package and compress (archive) files.

.\_ Run a command script in the current shell

!! Run the last command again

# Comment / Remark

// Comment / Remark