# ECGR 6181/8181 - Lab 0

**Objective**
This lab tutorial will introduce you to some of the basic Linux commands and steps to install Linux on VirtualBox, basic Linux commands, and software installation from source.

**Outcomes**
After this lab, you will be able to,
• Install VirtualBox
• Execute basic Linux commands
• Install a software from source on Linux

**Installation of Ubuntu Linux on Oracle VirtualBox**
**Note: You can skip this step if you are dual booting your laptop with Linux.**

Oracle VirtualBox is a cross-platform virtualization application. It installs on your existing Intel or AMD-based computers, whether they are running Windows, Mac, Linux or Solaris operating systems. VirtualBox can create and run a ”guest” operating system (virtual machine) in a window of the host operating system. The virtual machine provides a self-contained environment in which to experiment with new software without risking damaging changes to the host operating system.

Linux is available in various distributions (distros) which includes a Linux kernel, additional software, a windows system, and desktop environment. Examples of popular Linux distros include Ubuntu, Debian, Arch, Fedora, Mint, Manjaro, and Red Hat Enterprise Linux. We will be using Ubuntu in these labs. You are free to use another distribution if you wish.

1. Download Ubuntu 22.04 LTS latest stable version iso file from here, https://www.ubuntu.com/download/desktop
2. Go to VirtualBox website, https://www.virtualbox.org/wiki/Downloads to download the binary for your current operating system. When download is finished, run the executable file. Continue with the installation of VirtualBox

Follow instructions here to install Ubuntu Linux on Virtual box.
https://www.freecodecamp.org/news/how-to-install-ubuntu-with-oracle-virtualbox/

**Linux Command Line Interface (CLI)**
Linux has both, a graphical user interface (GUI) and a command line interface (CLI) that allow humans to interact with the computer. A command line interface is a type of human-computer interface that relies solely on textual input and output. A command line is a space on the display screen in which commands are typed in by the user. Pressing the ENTER key after typing in a command causes that command to be passed to the shell.

A shell, also referred to as a command interpreter, is a program that provides the CLI, reads commands that are typed by the user and executes them. The results of executing many but not all, commands are also shown on the command line.

The biggest advantage of the CLI with shells as compared with the GUI on Unix-like operating systems are power and flexibility. CLI programs often provide more options than their GUI counterparts. Also, multiple CLI commands can be combined using pipes to perform tasks that would be much more cumbersome to perform with GUI programs. Additionally the shell is programmable - that is you can write scripts to combine commands with programming constructs to automate tasks.

Yet another advantage of CLIs is that they are typically self-documenting. That is, they state exactly what the user wants to do, and more sophisticated CLIs keep a record of the commands that have been issued. This record can be accessed with the history command on Linux and other Unix-like operating systems. By keeping these advantages in mind, let's learn some basic Linux commands.

Linux supports many different types of shells such as Bash, Zsh, Csh, and Ksh.

Open a Linux terminal (clock of 9 dots icon on right button to launch applications; opens bash shell ) and try executing the following commands. $ indicates the command prompt as seen on the screen.
1. pwd (print working directory)
$ pwd
This command prints the absolute pathname of the working directory.

2. (list)
$ ls -l
The ls command lists the contents of your current working directory. The -l flag is for a long listing. Most Linux commands come with a number of flag options that modify the command output.

3. mkdir (make directory)
$ mkdir directory name
This command creates a new directory in the current working directory.

4. cd (change directory)
$ cd <sub directory name>
To change to a sub-directory inside the current working directory, this command can be used. Also cd .. allows to go to the parent directory of the current working directory.

5. cp (copy)
$ cp file1 file2

This command makes a copy of file1 in the current working directory and names it file2.

6. mv (move)
$ mv file1 file2
To move a file from one place to another, mv command can be used. This has the effect of moving rather than copying the file, so you end up with only one file rather than two.

7. rm (remove)
$ rm file.txt
To delete (remove) a file, use the rm command. To delete a directory you will need to use -r flag (for recursive)

8. clear (clear screen)
$ clear
This command helps to clear the terminal window of the previous commands and their output.

9. cat (concatenate)
$ cat file.t
This command can be used to display the contents of a file on the screen.

10. less
$ less file.txt
This command writes the contents of a file onto the screen single page at a time.

11. grep
The command grep has several usages. One of the usages is, it helps the user to search a specific word in a file.
Example usage:
$ grep <word> file.txt

12. man
Example usage
$ man ls
The man command is used to format and display the man pages. The man pages provide extensive documentation about commands and other aspects of the system, including configuration files, system calls, library routines and the kernel.

13. chmod (changing a file mode)
chmod is used to change the permissions of files or directories. The options for chmod are as follows.

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

For example, to remove read write and execute permissions of a file for the group and others, type
$ chmod go-rwx <file name>
See the link below for the numeric method as well
https://linuxize.com/post/chmod-command-in-linux/

14. su (substitute user)
$ su
The su command makes it possible to change a login session's owner without the owner having to first log out of that session.

15. ps (process status)
$ ps
The ps command is used to provide information about the currently running processes, including their process identification numbers (PIDs).

16. kill
$ kill <process id>
The kill command is used to terminate processes with process id obtained from ps

**Combining commands with pipe**

You can output the results of one command to the input of another commands using Linux pipes (represented by | sign). This helps you to chain commands to create a complex processing pipeline.

Let's use 3 commands - ps, grep, and awk to list the process IDs (PIDs) for all systemd-related processes. You have seen ps and grep before.  Awk is a text processing tool.

$ ps -ef | grep systemd | awk '{ print $2 }'
To gain more insight build the pipe a command at a time, that is, first ps -ef, then pipe it to grep systemd, and then pipe it to awk '{ print $2 }'
Do an online search to learn more about each command including the command line flags.

**Basics of software installation on Linux**

Repositories
A “repository” in this context is a public server hosting installable software packages. A Linux distribution provides a command, and usually a graphical interface to that command, that pulls the software from the server and installs it onto your computer. Most modern Linux distributions have standard repositories that include most of the software you’ll need to successfully run your Linux desktop. When a package is installed from a repository the installation is managed with the built-in package manager. This should be considered a best practice. Because it’s important for the integrity of the platform to ensure the package manager is aware of installed software. Also, packages can easily be updated in this case. Another reason to install from repositories is that dependencies are easily met. When installing from source, you have to make sure that all dependencies are met before the installation.

Since, repositories have become so inclusive, rarely you will ever need to install a package by any other means. However, if a package is not found in any repository or a it is developed in-house or a it has custom dependencies then you will have to install the package from the source.

APT is a tool, commonly used to install packages, remotely from the software repository. It’s a simple command-based tool that you use to install software.
1. To install a package you can use,
$ sudo apt-get install package-name
(Installing or removing any packages requires super user privileges.)

2. To remove a package you can use,
$ sudo apt-get remove package-name

3. To update the APT database, upgrade any security updates and patches that might be available for some installed software, you can use,
$ sudo apt-get update

To see more variations of apt-get command, you can visit the website,
https://itsfoss.com/apt-get-linux-guide/

**Software Installation from source**

In this example we will be compiling and installing a sample software called, curl from source. curl is a computer software project providing a library and command-line tool for transferring data using various protocols. The basics used in this example applies to the majority of packages and can be applied in most cases.

1. To compile sources on Linux, you will need the package called “build-essential” on Debian-based systems (such as Ubuntu). To install this on Ubuntu run,

$ sudo apt-get install build-essential
Some packages require you to have some dependencies installed in order to be compiled or to be run afterwards. When using apt or another package manager, it usually handles this for you. When compiling packages yourself, you should always check the documentation, and make sure you have the required packages installed beforehand

2. The first thing we need to download is the curl sourcecode. There are a lot of different ways to download the source, but in this example, we will use the tarball available from the curl website.
$ wget -O curl.tar.gz https://curl.se/download/curl-7.84.0.tar.gz

3. This will download and save the source as curl.tar.gz in your current directory. Next we will need to extract the tarball. To do this run,
$ tar -xvzf curl.tar.gz

4. Next go to the extracted tarball directory by typing,
$ cd curl-7.84.0

5. Inside the folder you will notice a lot of different files. For now, we will just be focusing on the file called “configure”. “configure” is a script designed to aid a program to be run on a wide number of different computers. Go ahead and run,
$ ./configure

This will automatically use your system variables to configure and ready the source for your machine. It basically matches the libraries required by the program, with the ones installed on your system. By doing this, the compiler knows where to look for the libraries required by the source, or in this case by curl. Besides that, it will also figure out where to install the package afterwards. When it is done it will generate a file called Makefile with all the info in it.

6. You are now ready to compile the source. To compile it run the command,
$ make

7. This will compile the source. It should take about a minute or so. When it is done, you should be ready to install it. Then run,
$ sudo make install

8. Make will now follow the instructions in the Makefile to install the compiled package. In most cases you should be done now. You can go ahead and type curl now. If curl has been installed properly you should see something like this,
curl: try ’curl –help’ or ’curl –manual’ for more information