LINUX

Part 1

What is Linux?

Ubuntu is built on the foundation of Linux, which is a member of the Unix family. Unix is one of the oldest types of operating systems, and together with Linux has provided reliability and security for professional applications for almost half a century. Many servers around the world that store data for popular websites (such as YouTube and Google) run some variant of Linux or Unix. The popular Android system for smartphones is a Linux variant; modern in-car computers usually run on Linux. Even the Mac is based on Unix. The Linux kernel is best described as the core—almost the brain—of the Ubuntu operating system. The Linux kernel is the controller of the operating system; it is responsible for allocating memory and processor time. It can also be thought of as the program which manages all applications on the computer itself.

Ubuntu Philosophy

The term “Ubuntu” is a traditional African concept originating from the Bantu languages of southern Africa. It can be described as a way of connecting with others—living in a global community where your actions affect all of humanity. Ubuntu is more than just an operating system: it is a community of people coming together voluntarily to collaborate on an international software project that aims to deliver the best possible user experience.

Part 2

This tutorial gives you an opportunity to try basic commands. By following the procedures in this section, you will learn to:

1. How to access Terminal
2. Obtaining Help
3. View the contents of a directory
4. Change from one directory to another
5. Create Directories
6. Delete Directories
7. Create File
8. Copy files
9. Rename files & Move Files
10. Delete files

1. How to access Terminal

Terminal will be under **Applications menu -> Accessories -> Terminal**

Or you can access it with shortcut key **ctrl+alt+T**

2. Obtaining Help

The "man" command

Almost every command in Linux has online help available from the command line, through the "man" (manual) command.

Try it now - type in "man ls". The resulting page will describe the command, then describe every option, then give further details about the program, the author, and so on. This information is shown using the "less" command (which we'll describe later on). For now, it is sufficient to know that you can use the up and down arrow, PgUp and PgDn keys to move around, and the Q key to quit.

The "info" command

Another source of online help is the "info" command. Some Linux commands may supply both "man" and "info" documentation. As a general rule, "info" documentation is more verbose and descriptive, like a user guide, while "man" documentation is more like a reference manual, giving lists of options and parameters, and the meaning of each.

Try typing "info ls" now. The method for moving around in "info" is quite similar to "man" - you can also use the arrows and PgUp/PgDn to move, and Q to quit. The main difference is that info

pages can contain "menus" of links which lead to other pages. To follow a link, move the text cursor to it with the arrow keys, and press Enter.

The "--help" option

Most (but not all) programs have a --help option which displays a very short description of its main options and parameters. Try typing "ls --help" to see. This will produce more than one screenful of information, so you'll have to use the terminal's scrollbar to see what was displayed.

The "--help" information rarely says anything that isn't also found in the "man" documentation, so it's rarely needed, except in a tiny number of programs which do not supply any other form of documentation.

3. View the contents of a directory

ls

The **ls** command outputs a list of the files in the current directory. When used from a terminal, it generally uses colors to differentiate between directories, images, executable files etc. As you can see, the prompt reappears at the end.

For example, typing and press Enter

```
ls
```

will show you the files that are in your home directory.

Adding options

Like practically all commands in Linux, you can add options to the "ls" command to alter its output or influence its behaviour. An option is preceded by a dash (eg, "ls -a") Try out the following variations of the ls command, to see different forms of output:

ls -l

Produces a "long format" directory listing. For each file or directory, it also shows the owner, group, size, date modified and permissions

ls -a

Lists *all* the files in the directory, including hidden ones. In Linux, files that start with a period (.) are usually not shown.

ls -R

Lists the contents of each subdirectory, their subdirectories etc (recursive).

4. Change from one directory to another

cd

The **cd** command changes directories. When you open a terminal you will be in your home directory. To move around the file system you will use **cd**.

To navigate into the root directory, type:

```
cd /
```

No matter which directory you are in, this command always returns you to the root directory of a drive. The root directory does not have a name. It is simply referred to by a backslash (\).

To navigate up one directory level, type:

```
cd ..
```

5. Create Directories

mkdir

The **mkdir** command allows you to create directories. For example, typing:

```
mkdir music
```

will create a directory named `music` in the current directory.

6. Delete Directories

rmdir

The **rmdir** command allows you to delete directories. For example, typing:

```
rmdir music
```

As a safety measure, the directory must be empty before it can be deleted.

5. Create Files

vi

The **vi** command allows you to create a new file. For example, typing:

```
vi match.doc
```

will create a file named `match.doc` in the current directory.

If in your environment, you don't have the package, you may update your app and install vim.

Proceed with the following command first:

Update the package list:

```
sudo apt update
```

Install "vim"

```
sudo apt install vim
```

To exit the file:

1. Press "**Esc**" to ensure you are in Normal mode.
2. Type "**:q!**" and press Enter

To save and exit:

1. Press "**Esc**" to ensure you are in Normal mode.
2. Type "**:wq**" and press Enter

less

The **less** command allows you to view the contents of file.

```
less match.doc
```

press "*q*" to quit.

6. Copy files

cp

When you use the copy command, you must include two parameters. The first is the location and name of the file you want to copy, or the source. The second is the location to which you want to copy the file, or the destination. You separate the source and destination with a space.

The **cp** command allows you to make a copy of a file.

```
cp source_file target_directory
```

7. Rename files & Move Files

mv

The **mv** command allows you to move a file to a different location or rename a file.

When use the **mv** command to rename a file, you must include two parameters.

The first is the file you want to rename, and the second is the new name for the file. You separate the two names with a space.

Examples are as follows:

```
mv match.doc game.doc
```

name of match.doc has changed to game.doc

When using **mv** command to move a file, two parameters are needed. The first is the location and name of the file you want to move, or the source. The second is the location to which you want to move the file, or the destination. You separate the source and destination with a space.

```
mv source_file target_directory/
```

Example:

Let assume you are currently located in database folder in which you want to move a file (chap1.ppt) from current location into another directory called “os”. Both directories are at the same level

```
mv chap1.ppt ../os/
```

8. Delete files

rm

The **rm** command allows you remove a file, for example:

```
rm game.doc
```