Table of Contents

Introduction to Linux..1

Ubuntu Linux 6

Ubuntu Operating System 6

Preparing to Install Ubuntu 7

Install Ubuntu 10

Linux Terminal 15

How to Deal with Command Line...15

Search Command 21

Commands Related to Dealing with Files and Folders 23

General Commands...23

Working with Files 31

Working with Folders 41

Copy, Move, Delete and Rename Commands...42

Copy and Move with a tag * 47

Compression and Archiving..49

Archiving...49

Compression..52

Reference 53

Abbreviation List

ABBREVIATION	MEANING
OS	Operating System
WSL	Windows Subsystem for Linux
LTS	Long Term Support

Introduction to Book

Linux is a family of open source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on **September 17, 1991**, by **Linus Torvalds**. **Linux** is typically packaged in a **Linux** distribution.

Distributions include the **Linux kernel** and supporting system software and libraries, many of which are provided by the **GNU** Project. Many Linux distributions use the word "Linux" in their name, but the Free Software Foundation uses the name **GNU/Linux** to emphasize the importance of GNU software, causing some controversy.

Popular Linux distributions include **Debian**, **Fedora**, and **Ubuntu**. Commercial distributions include Red Hat Enterprise Linux and SUSE Linux Enterprise Server. Desktop Linux distributions include a windowing system such as **X11** or **Wayland**, and a desktop environment such as **GNOME** or **KDE** Plasma. Distributions intended for servers may omit graphics altogether, or include a solution stack such as **LAMP**. Because **Linux** is freely redistributable, anyone may create a distribution for any purpose.

Linux was originally developed for personal computers based on the **Intel x86** architecture, but has since been ported to more platforms than any other operating system Linux is the leading operating system on servers and other big iron systems such as mainframe computers, and the only **OS** used on **TOP500** supercomputers (since November 2017, having gradually eliminated all competitors).It is used by around 2.3 percent of desktop computers. The Chromebook, which runs the Linux kernel-based **Chrome OS**, dominates the **US K–12** education market and represents nearly **20 percent** Develop a good working knowledge of **Linux** using both the graphical interface and command line, covering the major Linux distribution families.

Linux powers 100% of the world's supercomputers, most of the servers powering the Internet, the majority of financial trades worldwide and over two billion Android devices. In short, Linux is everywhere. It appears in many different architectures, from mainframes to server to desktop to mobile and on a staggeringly wide variety of hardware.

Moreover, 80 percent of hiring managers reported that they will prioritize hiring Linux talent relative to other skills areas, and **47 percent** of hiring managers say they’re more likely to hire a candidate with Linux certification.

In this book we will only learn the basics of Linux in order to start working with Linux later.

What you'll learn

In the **first chapter**, we will take an introduction to **Linux**.

The **second chapter** will explain about a version of Linux that is the most widely used, and it is **Ubuntu.** Only the installation method will be explained, after which you can start learning to use Linux.

Chapter three will explain a set of commands for Terminal window.

CHAPTER ONE
INTRODUCTION TO LINUX

Eng. Elaf A.Saeed

Chapter One
Introduction to Linux

Linux is a community of **open-source Unix** like operating systems that are based on the **Linux Kernel**. It initially released by Linus Torvalds on September 17, 1991. It is a **free** and **open-source** operating system and the source code can be modified and distributed to anyone commercially or non-commercially under the GNU General Public License. Initially, **Linux** is created for personal computers gradually it is used in other machines like servers, mainframe computers, supercomputers, etc. Nowadays, Linux is also used in embedded systems like routers, automation controls, televisions, digital video recorders, video game consoles, smartwatches, etc. The biggest success of Linux is **Android** (operating system) it is based on the **Linux kernel** that is running on smartphones and tablets. Due to android Linux has the largest installed base of all general-purpose operating systems. Linux is generally packaged in a Linux distribution.

Linux Distribution

Linux distribution is an operating system that is made up of a collection of software based on **Linux kernel** or you can say distribution contains the Linux kernel and supporting libraries and software. And you can get Linux based operating system by downloading one of the Linux distributions and these distributions are available for different types of devices like embedded devices, personal computers, etc. Around 600 + Linux Distributions are available and some of the popular Linux distributions are:

- MX Linux
- Manjaro
- Linux Mint
- elementary
- Ubuntu
- Debian
- Solus

- Fedora
- openSUSE
- Deepin

Architecture of Linux

Linux architecture has the following components in figure 1.1:

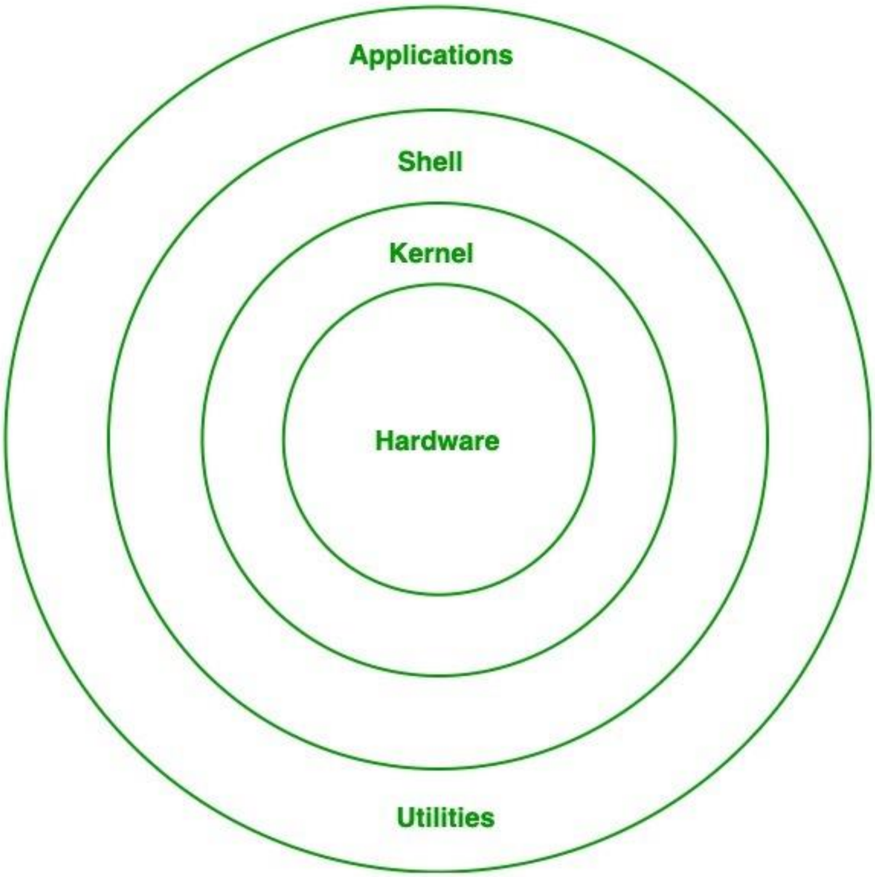


Figure 1.1: Architecture of Linux

1. **Kernel:** Kernel is the core of the Linux based operating system. It virtualizes the common hardware resources of the computer to provide each process with its virtual resources. This makes the process seem as it is the sole process running on the machine. The kernel is also responsible for preventing and mitigating conflicts between different processes. Different types of the kernel are:

- Monolithic Kernel

- Hybrid kernels
- Exo kernels
- Micro kernels

2. **System Library:** It is the special types of functions that are used to implement the functionality of the operating system.
3. **Shell:** It is an interface to the kernel which hides the complexity of the kernel's functions from the users. It takes commands from the user and executes the kernel's functions.
4. **Hardware Layer:** This layer consists all peripheral devices like RAM/ HDD/ CPU etc.
5. **System Utility:** It provides the functionalities of an operating system to the user.

Advantages of Linux

- The main advantage of Linux, is it is an open-source operating system, means the source code is easily available for everyone and you are allowed to contribute, modify and distribute the code to anyone without any permissions.
- In terms of security, Linux is more secure than any other operating system. It does not mean that Linux is 100 percent secure it has some malware for it but is less vulnerable than any other operating system. So, it does not require any anti-virus software.
- The software updates in Linux are easy and frequent.
- Various Linux distributions are available so that you can use them according to your requirements or according to your taste.
- Linux is freely available to use on the internet.
- It has large community support.
- It provides high stability. It rarely slows down or freezes and there is no need to reboot it after a short time.
- It maintains the privacy of the user.
- The performance of the Linux system is much higher than other operating systems. It allows a large number of people to work at the same time and it handles them efficiently.

- It is network friendly.
- The flexibility of Linux is high. There is no need to install a complete Linux suit you are allowed to install only required components.
- Linux is compatible with a large number of file formats.
- It is fast and easy to install from the web. It can also install in any hardware even in your old computer system.
- It performs all tasks properly even if it has limited space on the hard disk.

Disadvantages of Linux

- It is not much user-friendly. So, it may be confusing for beginners.
- It has small peripheral hardware drivers as compared to windows.

How does Linux differ from other operating systems?

In many ways, Linux is similar to other operating systems you may have used before, such as Windows, OS X, or iOS. Like other operating systems, Linux has a graphical interface, and types of software you are accustomed to using on other operating systems, such as word processing applications, have Linux equivalents. In many cases, the software's creator may have made a Linux version of the same program you use on other systems. If you can use a computer or other electronic device, you can use Linux.

But Linux also is different from other operating systems in many important ways. First, and perhaps most importantly, Linux is open source software. The code used to create Linux is free and available to the public to view, edit, and—for users with the appropriate skills—to contribute to.

Linux is also different in that, although the core pieces of the Linux operating system are generally common, there are many distributions of Linux, which include different software options. This means that Linux is incredibly customizable, because not just applications, such as word processors and web browsers, can be swapped out. Linux users also can choose core components, such as which system displays graphics, and other user-interface components.

What is the difference between Unix and Linux?

You may have heard of Unix, which is an operating system developed in the 1970s at Bell Labs by Ken Thompson, Dennis Ritchie, and others. Unix and Linux are similar in many ways, and in fact, Linux was originally created to be similar to Unix. Both have similar tools for interfacing with the systems, programming tools, filesystem layouts, and other key components. However, Unix is not free. Over the years, a number of different operating systems have been created that attempted to be “unix-like” or “unix-compatible,” but Linux has been the most successful, far surpassing its predecessors in popularity.

Who uses Linux?

You’re probably already using Linux, whether you know it or not. Depending on which user survey you look at, between one- and two-thirds of the webpages on the Internet are generated by servers running Linux.

Companies and individuals choose Linux for their servers because it is secure, flexible, and you can receive excellent support from a large community of users, in addition to companies like Canonical, SUSE, and Red Hat, which offer commercial support.

Many of the devices you own probably, such as Android phones and Chromebooks, digital storage devices, personal video recorders, cameras, wearables, and more, also run Linux. Your car has Linux running under the hood. Even Microsoft Windows features Linux components, as part of the Windows Subsystem for Linux (WSL).

CHAPTER TWO
UBUNTU LINUX

Eng. Elaf A.Saeed

Chapter Two
Ubuntu Linux

In this chapter, I will explain about a type of **Linux** program, which is **Ubuntu**. The reason for choosing this version is that it is possible for the user who wants to use **Linux** to work with it for public use, it can be used by all users in general, and who wants to learn the Linux. Ubuntu is one of the simplest versions.

Therefore, we will explain how to install this version, through which you can learn to use it if you want to start using the Linux operating system.

Ubuntu Operating System

Ubuntu is currently an integrated system and is one of the most widespread distribution of **Linux** due to its focus on the end user and simplification of things in addition to attention to aesthetics and the provision of the system in all languages of the world, and all this without prejudice to the security of the system, the first release of **Ubuntu** was in **2004**, Each year two versions are released.

The system issues a regular copy and a long-term copy. The support period for the regular version is one and a half years. As for the long-term version, which is denoted by **LTS** (Long Term Support), it is issued every two years and supports for three years for desktop devices and five years for servers. This version is very practical in productive places and the business sector.

Ubuntu features:

1- Ease of trial and installation.

2- Ease of use.

3- No viruses.

4- Light, sophisticated, and stable.

5- Beautiful and elegant.

6- Convenience and versatility.

7- A large community.

8- Open source, free.

Ubuntu operating requirements

In general, **Ubuntu** does not need a computer with great specifications, it can work on any device with medium capabilities, and even few in light of the developments taking place in computers these days, and we now in table 2.1 know the minimum capabilities that **Ubuntu** needs in addition to the recommended capabilities to get the best performance.

Table 2.1: the Minimum Capabilities the Ubuntu Needs

	Processor	*Memory*	*Hard*	*Others*
Minimum	**1 GB**	**512 MB**	**8 GB**	**Network-sound card**
Recommended	**8 GB**	**1.5 GB**	**15 GB**	**Network-sound card**

Preparing to Install Ubuntu

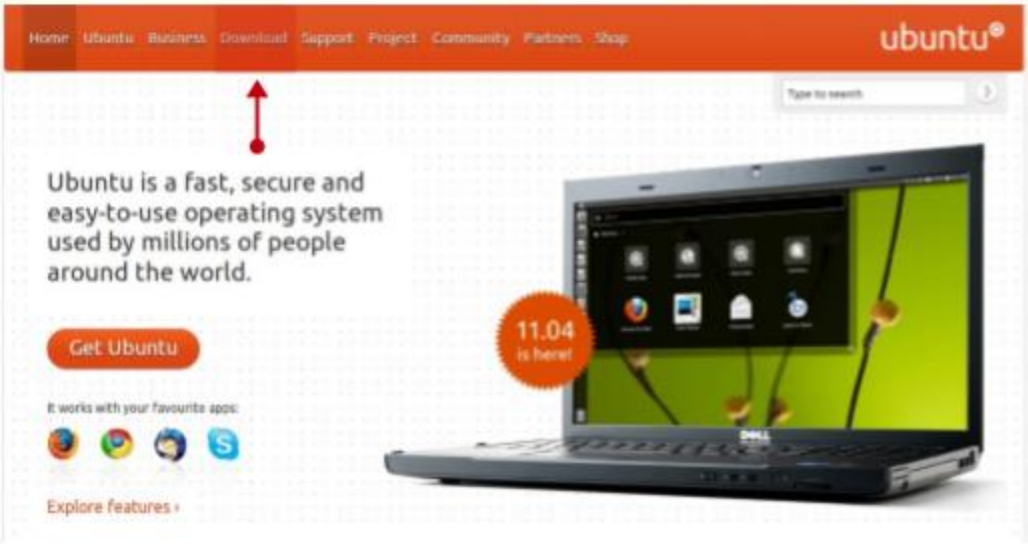
Download Ubuntu

Ubuntu is a free open source system and also free, so you can get it from the internet, from your friend, CDs, and that will not be considered a violation of the law as in other systems. The easiest way to get **Ubuntu** releases downloaded from the **Internet**, **burn them to a CD or work on a USB flash memory**.

We will now know the steps to download the latest version of Ubuntu from the official website.

1- Now go to the official website of Ubuntu, and choose **Download**, as shown in figure 2.1. **https://www.ubuntu.com/**

Chapter Two
Ubuntu Linux

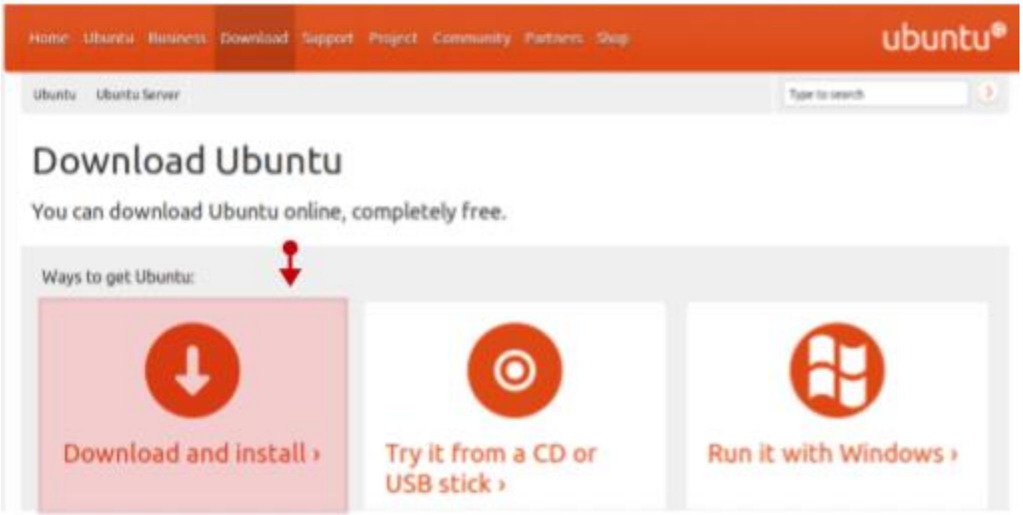


Ubuntu is a fast, secure and easy-to-use operating system used by millions of people around the world.

It works with your favourite apps:

Figure 2.1: The official website of the Ubuntu system

2- After that you will see the next page choose **Download and install**, as shown in figure 2.2.



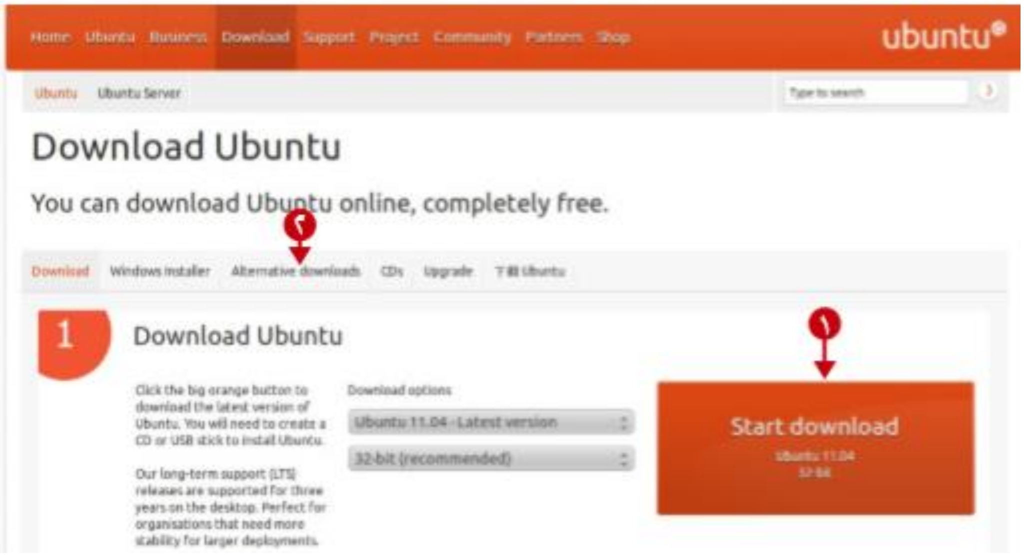
Download Ubuntu

You can download Ubuntu online, completely free.

Ways to get Ubuntu:

Figure 2.2: The Ubuntu download page shows several download options

3- You get to the download page that shown in figure 2.3.



Download Ubuntu

You can download Ubuntu online, completely free.

Download Ubuntu

Click the big orange button to download the latest version of Ubuntu. You will need to create a CD or USB stick to install Ubuntu.

Our long-term support (LTS) releases are supported for three years on the desktop. Perfect for organisations that need more stability for larger deployments.

Figure 2.3: The Ubuntu download page shows several download options

1- Click **Start Download** to download directly.

2- Click on **Alternative Downloads** to get additional downloading methods. For example, choose **Bit Torrent download** to download via torrent.

❖ If you choose to download directly, it will start downloading the latest versions of Ubuntu in the size of a regular cylinder disc. It is always advised not to use violent download programs such as **IDM**, as these programs often damage the **ISO** file. And if you encounter problems with downloading the file intact, you prefer to download via **torrent technology**, as it is the best way to download **Ubuntu** properly.

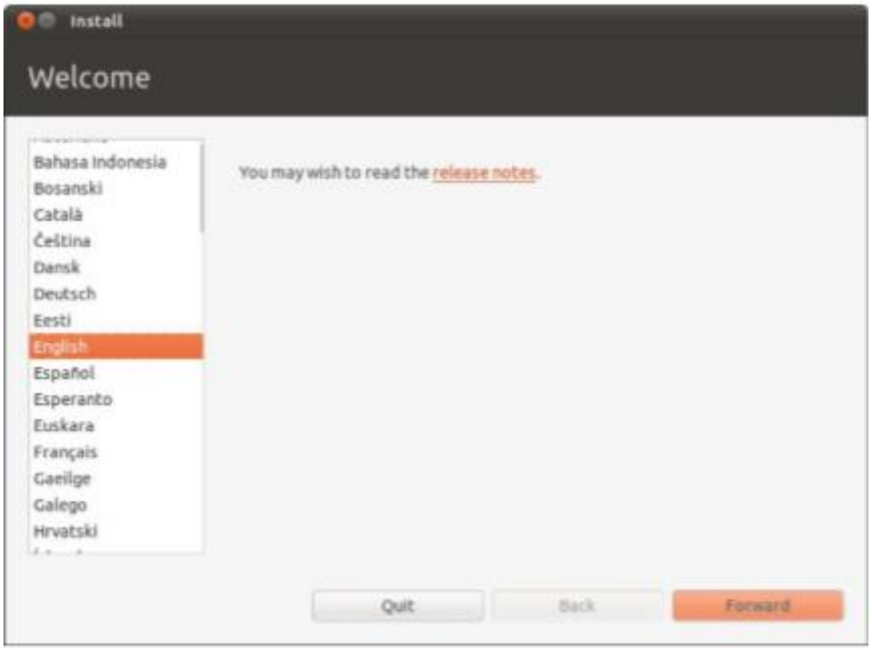
Customize a Partition for Ubuntu

The last step before installation is to allocate the partition on which the system will be installed. You will not need to use hard disk partition programs such as Partition Magic or similar software. Ubuntu will do all the work automatically as the system installs to the largest partition in free space.

Ubuntu will not use it will not use the entire free space but you will decide how much space the system will use, which, as we said earlier, is preferably 10 GB. If you want to specify the partition that will be installed on, all you have to do is transfer files from it to any other partition so that it is larger.

Install Ubuntu

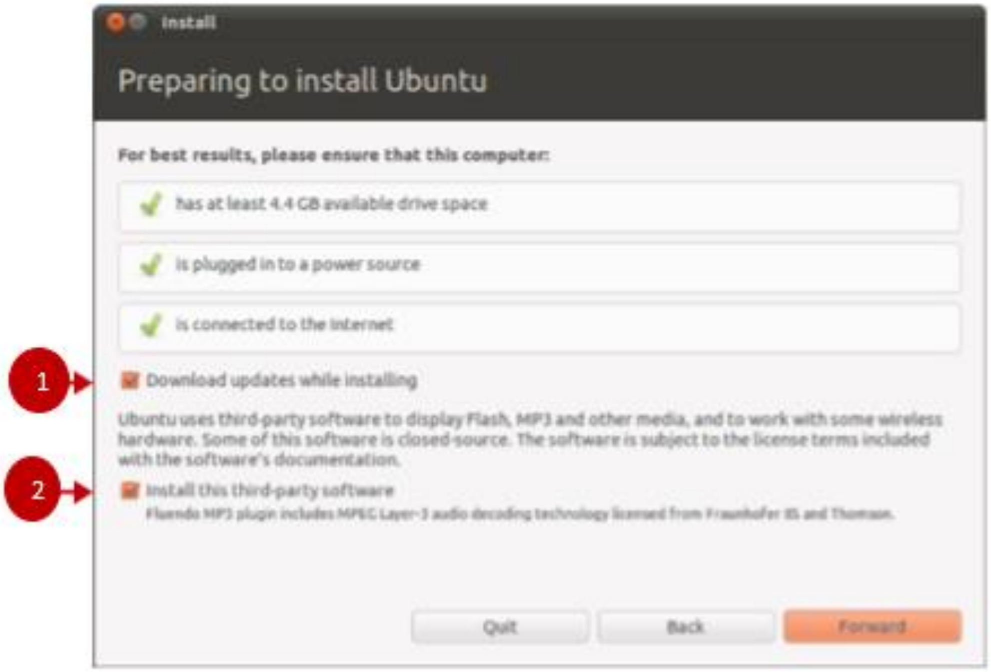
1- The first window that appears for you is to choose the interface language, which is English by default, as shown in figure 2.4.



You may wish to read the release notes.

Figure 2.4: Choose the Interface language

2- To get the perfect installation process, you must make sure that the installation requirements are met, make sure that there is a hard disk space of no less than **4.4 GB**, and that the device is connected to electricity if you are installing on a mobile device (laptop), and finally there is an Internet connection. In figure 2.5 that shown the preparing to install ubuntu window.



Install

Preparing to install Ubuntu

For best results, please ensure that this computer:

✓ has at least 4.4 GB available drive space

✓ is plugged in to a power source

✓ is connected to the internet

☑ Download updates while installing

Ubuntu uses third-party software to display Flash, MP3 and other media, and to work with some wireless hardware. Some of this software is closed-source. The software is subject to the license terms included with the software's documentation.

☑ Install this third-party software

Fluendo MP3 plugin includes MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and Thomson.

Figure 2.5: Prepare for the Ubuntu installation process with the ability to download updates during installation

For the other two options, they are as follows:

1- When this option is checked, the system will download the updates while installing.

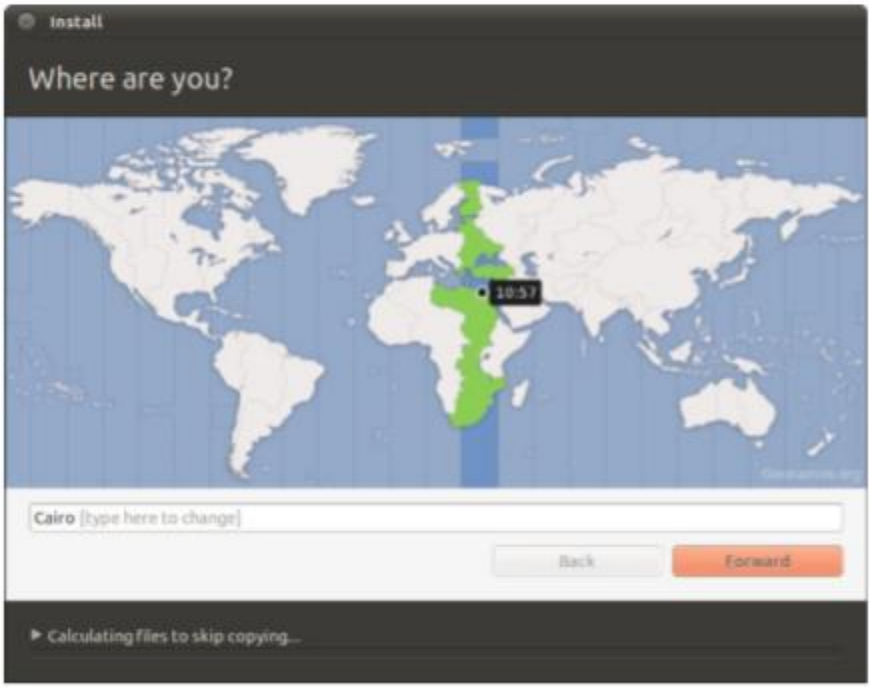
2- When checking this option, the system will download some restricted software to play some formats, such as MP3, and thus Ubuntu will be able to play basic audio formats.

3- After that a window appears to choose how to install Ubuntu, for example do you want to install Ubuntu alongside another system like Windows? Or do you want Ubuntu only, without any other system on the hard drive? In general, it is preferred that you install Ubuntu next to the other system that you use until you get used to it, and then decide whether you want to keep the other system or not?

4- The next step is to determine the size of the Ubuntu partition, and Ubuntu provides a very easy way to do this without any complications.

Ubuntu will search for the largest partition of the hard drive name with an empty space and takes part of it to install Ubuntu in it and all you have to do is define the area of the Ubuntu partition with the mouse pointer!

5- We come to the step of locating you on the map to set the time, as shown in figure 2.6.



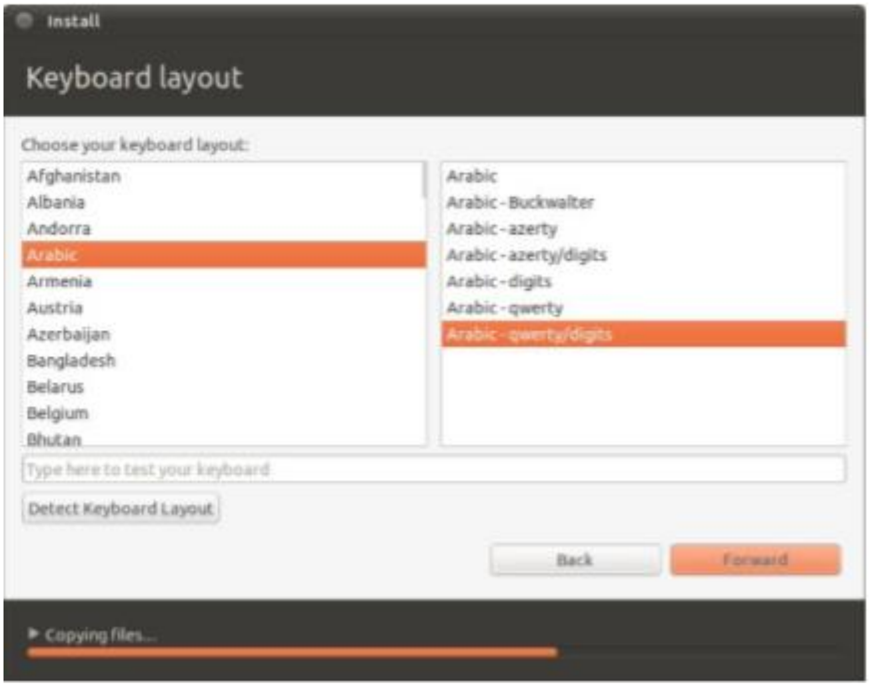
Where are you?

Cairo [type here to change]

▸ Calculating files to skip copying...

Figure 2.6: locating you on the map

6- Now the step of choosing the language of the keyboard, as shown in figure 2.7.



Keyboard layout

Choose your keyboard layout:

- Afghanistan
- Albania
- Andorra
- Arabic
- Armenia
- Austria
- Azerbaijan
- Bangladesh
- Belarus
- Belgium
- Bhutan

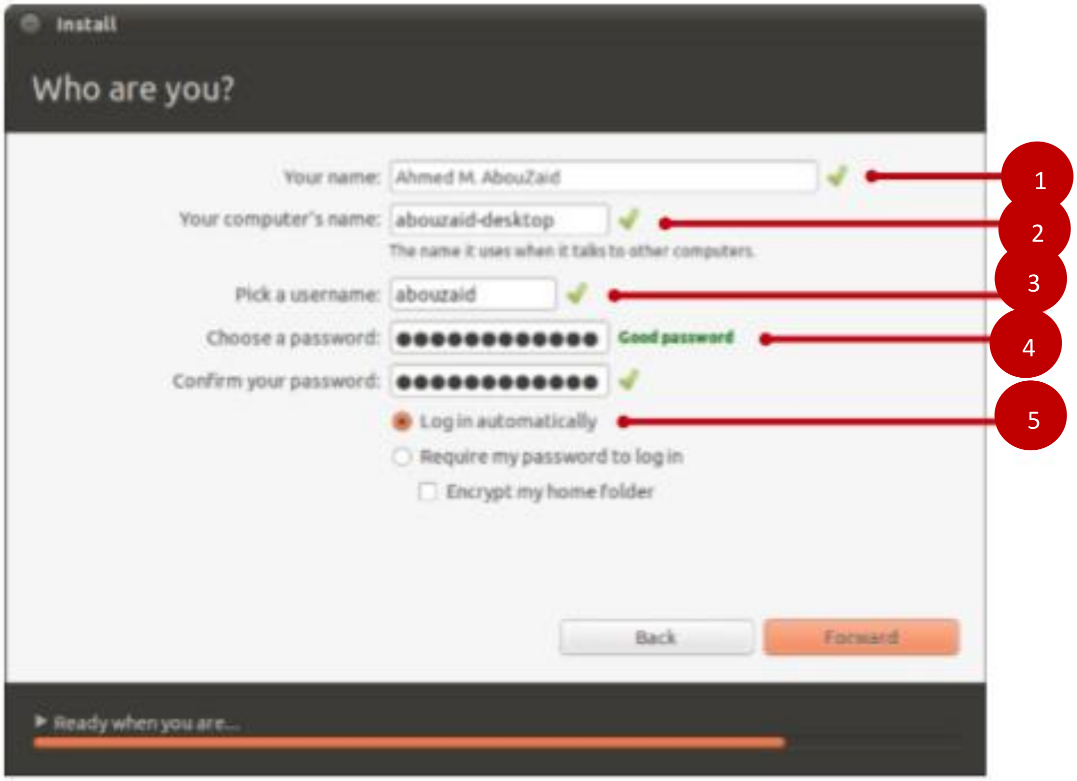
- Arabic
- Arabic - Buckwalter
- Arabic - azerty
- Arabic - azerty/digits
- Arabic - digits
- Arabic - qwerty
- Arabic - qwerty/digits

Type here to test your keyboard

▸ Copying files...

Figure 2.7: Keyboard Language

7- The current step is to choose a username and password, as shown in figure 2.8.



Install

Who are you?

Your name: Ahmed M. AbouZaid ✓

Your computer's name: abouzaid-desktop ✓

The name it uses when it talks to other computers.

Pick a username: abouzaid ✓

Choose a password: ●●●●●●●●●●●●● Good password

Confirm your password: ●●●●●●●●●●●● ✓

● Log in automatically

○ Require my password to log in

□ Encrypt my home folder

▶ Ready when you are...

Figure 2.8: Prepare user data

1- Your name.

2- The name of the device.

3- Your username.

4- The user's password.

5- Make sure to choose this option.

8- If you have more than one operating system on the device, such as Windows, Ubuntu can import your data from other systems, thus saving you the trouble of transferring it manually, and Ubuntu becomes ready for use quickly.

Choose the data you want to import, for example the browser data, such as the sites you visited, the preferred sites, and other data that makes your use of the Ubuntu system smoother.

Now the process to install Ubuntu begins.

In figure 2.9 that shown the Desktop interface after installing Ubuntu.

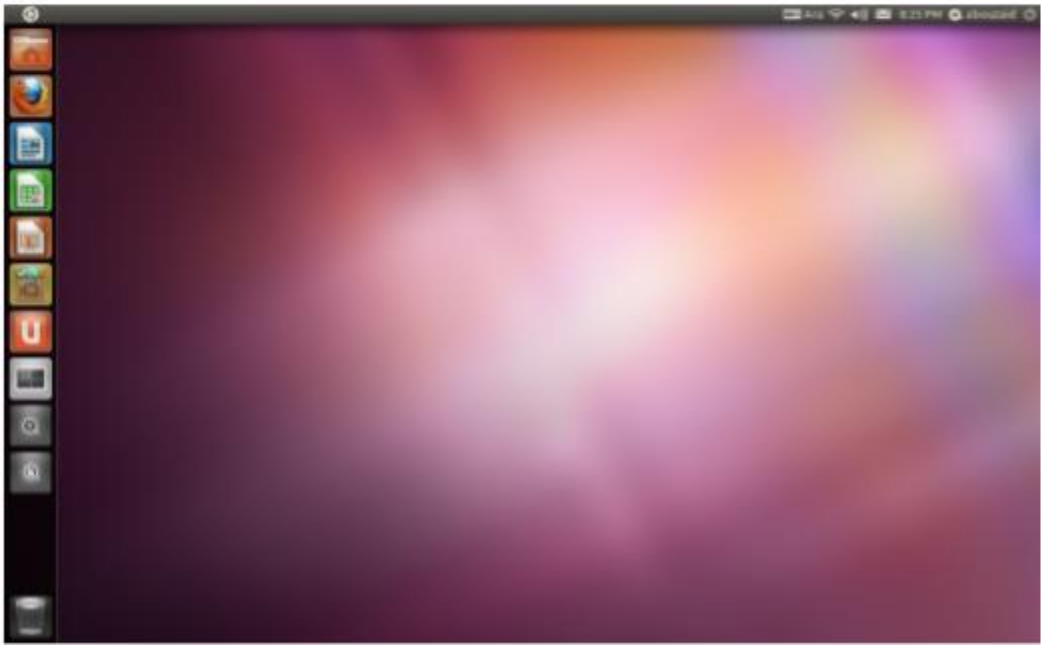


Figure 2.9: Desktop interface after installing Ubuntu

CHAPTER THREE
LINUX TERMINAL

Eng. Elaf A.Saeed

Chapter Three
Linux Terminal

How to deal with the command line

In order to understand the task of one of the commands and its multiple uses and options, you can use these commands to be able to surround the command's function.

# For each command, there are manual pages that help define the bitter function and show how to use it.

- **This sign 🟥 means a space.**

1- man

Command function: shows you all the information related to any command you want, such as (how to use this command - the name of the programmer who wrote this command and his email - other relevant commands in this matter .... etc.). This command is used as follows:

man🟥 The command you want to know about it.

1- If you want to know all information related to command ls, write the following:

```
elaf@the-game:~> man ls
```

2- To know the function of a specific command without full display of the help page, we use the (f) option. For example, if you want to know the function of ls, write the following:

```
elaf@the-game:~> man -f ls
```

The answer is as follows:

```
ls (1p)                                      - list directory contents
 ls ( 1)                                      - list directory contents
```

That is, the ls command is used to display the contents of folders

3- To find out all the commands that perform a specific purpose, use the (k) option. For example, if you want to know all the commands that make fdisk, write the following:

```
elaf@the- game:~> man -k fdisk
```

The answer is as follows:

```
cfdisk (8)              - Curses based disk partition table
                          manipulator for Linux
 fdisk (8)              - Partition table manipulator for Linux
sfdisk (8)              - Partition table manipulator for Linux
```

Note: The options [-f] and [-k] depend on **whatis** command database.

In table 3.1 that shown the keys for handling manual help pages:

Table 3.1: keys for handling manual help pages

Down arrow	To go to the end of the page line after line.
Up arrow	To go to the beginning of the page line after line.
Page down	To move a page down.
Page up	To move a page after page up.
q	To exit the manual page.

2- info:

This command has the same function as the previous command man, but the info command is more detailed.

This command is used as follows:

info█ The command you want to know about it.

1- If you want to know all the information related to the command **ls**, write the following:

```
elaf@the-game:~> info ls
```

In table 3.2 that shown the keys for dealing with help pages info:

Table 3.2: keys for dealing with help pages info

e	To go to the end of the page.
b	To go to the beginning of the page.
Down arrow	Go to the last page step by step.
Up arrow	Go to the beginning of the page, step by step.
*****	When this flag is present, it indicates the presence of a new page, move with the arrow, pause on it, and press Enter to enter it the page.
Shift + ?	Find out useful keys on the info page.
d	Returns you to the main page of the info.
q	To exit the info page.

3- xman:

This command is used to display **manual help pages** just like the **man** command, but it is distinguished by the following (you can use the mouse to view and search orders - it can display commands according to their number (1 - 9) as it will be explained in the section how to understand man pages).

This command is used as follows:

```
elaf@the-game:~> xman &
```

You will see the following figure 3.1:

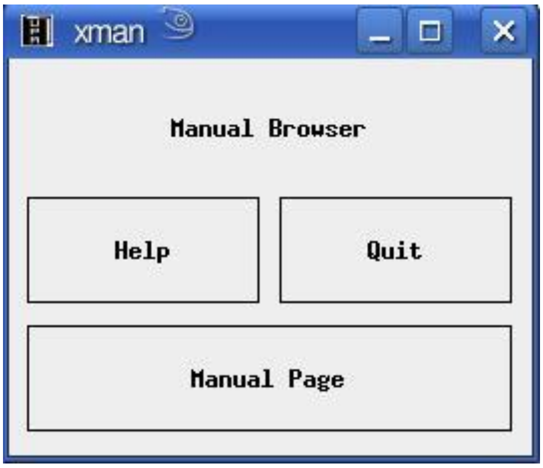
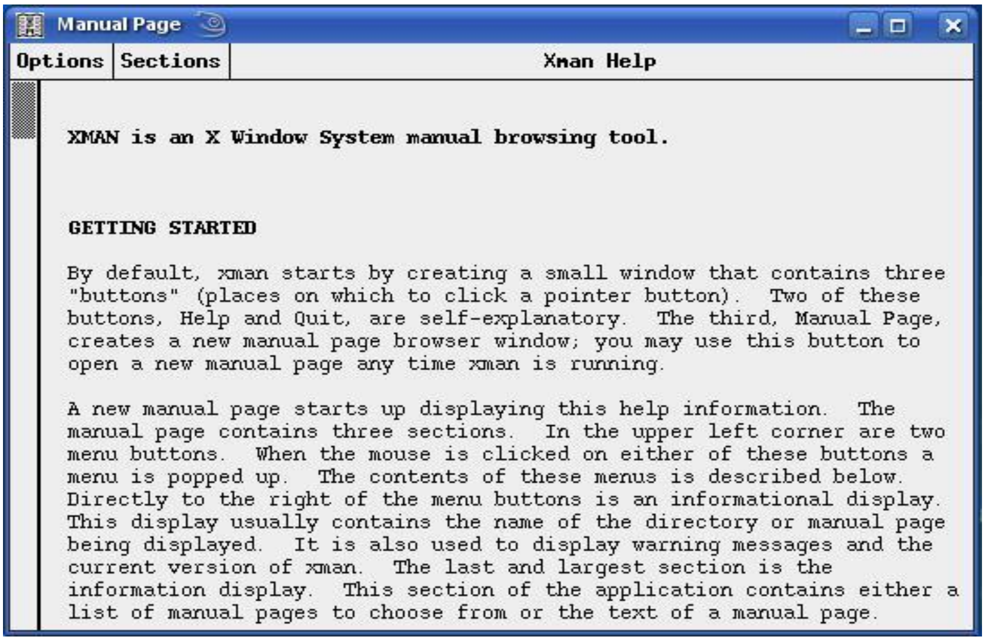


Figure 3.1: xman Window

To view help pages, click on the manual page and it will show you the figure 3.2:



```
XMAN is an X Window System manual browsing tool.


GETTING STARTED

By default, xman starts by creating a small window that contains three
"buttons" (places on which to click a pointer button).  Two of these
buttons, Help and Quit, are self-explanatory.  The third, Manual Page,
creates a new manual page browser window; you may use this button to
open a new manual page any time xman is running.

A new manual page starts up displaying this help information.  The
manual page contains three sections.  In the upper left corner are two
menu buttons.  When the mouse is clicked on either of these buttons a
menu is popped up.  The contents of these menus is described below.
Directly to the right of the menu buttons is an informational display.
This display usually contains the name of the directory or manual page
being displayed.  It is also used to display warning messages and the
current version of xman.  The last and largest section is the
information display.  This section of the application contains either a
list of manual pages to choose from or the text of a manual page.
```

Figure 3.2: Manual Page

To display orders by their number, press the **sections** button.

To search for a specific command, click on the **options** button, then click on **Search**, and you will see the following figure 3.3:

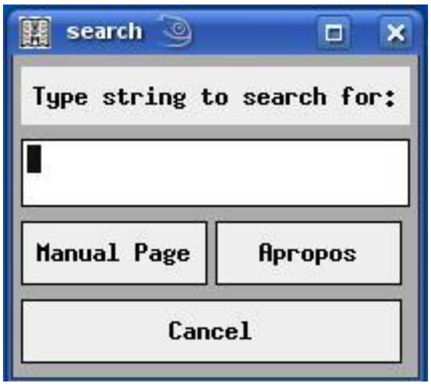


Figure 3.3: Search Page

To search for orders by name, press the **manual page** after entering the command name.

To search for orders by function, press **Apropos** after entering the command function.

4- whatis:

Used to display brief information about any command. In order for this to work, a database is required.

To make the database, use one of the following methods:

```
elaf@the-game:~> su
Password:
the-game:/home/elaf # makewhatis
```

If that method did not work for you, follow the second method:

```
elaf@the-game:~> su
Password:
the-game:/home/elaf # mandb
```

Now most distributions have a database ready for this matter and you do not need all of this, and it may not even have the command **makewhatis** at all.

The way to use the command is as follows:

whatis 🟥 The command you want to know about this job.

1- If you want to inquire about a job, just one command, write the following:

```
elaf@the-game:~> whatis ls
```

The result will be as follows:

```
ls (1)                         - list directory contents
```

2- If you want to inquire about a job with more than one thing, write the orders you want to inquire about, as follows:

```
elaf@the-game:~> whatis ls dir cd ifconfig
```

The result will be as follows:

```
      ls ( 1)                             - list directory contents
      dir (1)                             - list directory contents
       cd: nothing appropriate.
 ifconfig (8)                    - configure a network interface
```

We notice here that the **cd** command did not find a suitable job.

5- apropos:

Used to search for anything I know his job. That is, you use this command if you are looking for commands that perform a certain function.

The way to use the command is as follows:

apropos The command you want to know about it.

If you want to inquire about orders that can divide the hard disk, write the following:

```
elaf@the-game:~> apropos fdisk
```

The result will be as follows:

```
   cfdisk (8)                    - Curses based disk partition table
                                                manipulator for Linux
    fdisk (8)                  - Partition table manipulator for Linux
   sfdisk (8)                 - Partition table manipulator for Linux
```

The apropos command uses the same database as **whatis**.

Search Commands

1- whereis:

This command is used to search for programs [binary files] and related files and manual pages.

The way to use this command is as follows:

1- If you want to search for a specific program or command and everything related to it (such as its manual pages and source files), write the following:

whereis 🟥 The Program Name

Example: to search for the command ls and everything related to it.

```
elaf@the-game:~> whereis ls
```

The result will be as follows:

```
ls: /bin/ls /usr/share/man/man1p/ls.1p.gz
/usr/share/man/man1/ls.1.gz
```

whereis🟥-m 🟥 The Program Name

Example: To search for manual pages for the ls command only.

```
elaf@the-game:~> whereis -m ls
```

The result will be as follows:

```
 ls: /usr/share/man/man1p/ls.1p.gz
 /usr/share/man/man1/ls.1.gz
```

3- If you want to search for binary files, just write the following:

whereis🟥-b 🟥The Program Name

Example: To search for binary files for the ls command only:

```
elaf@the- game:~> whereis -b ls
```

The result will be as follows:

```
ls: /bin/ls
```

4- If you want to search for the original files {source}, just write the following:

whereis🟥-s 🟥 The Program Name

Example: To search for the original {source} files for the **ls** command only.

```
elaf@the- game:~> whereis -s ls
```

The result will be as follows:

```
ls:
```

2- locate:

It is used to search for files inside the system or inside the hard disk, and it is faster than the find command.

Note: Some distributions need to make a database for the locate command before using it, as follows:

```
elaf@the-game:~> su


Password:
the-game:/home/elaf # updatedb
```

The way to use this command is as follows:

locate🟥The file to search for

1- If you want to search for all files of one type only within your device, type the following:

locate *. The extension you want.

Example: To search for files with the .mp4 extension only within your device.

```
the-game:/home/elaf # locate *.mp4
```

2- If you want to search for files with the extension mp4 and mp2 within your device, type the following:

```
the-game:/home/elaf # locate *.mp4 *.mp2
```

3- If you want to search for a specific extension and let **avi** be **in / mnt / collection1** type the following:

```
elaf@the-game:~> locate /mnt/collection1/*.avi
```

Note: **{/mnt/collection1/*.avi}** write without spaces

Chapter Three
Linux Terminal

Commands Related to Dealing with Files and Folders

A - General Commands:

1- pushed:

Used to enter any folder or path. The way to use this command is as follows:

pushed🟥 The path to move to it.

Or

pushed🟥 The folder to move to it.

Example: If you want to go to the this track **{/ tmnt / games / chm}** type the following:

```
elaf@the-game:~> pushd /mnt/games/chm/
```

Example: If you want to move to the folder {elaf}, type the following:

```
elaf@the-game:/mnt/games/chm> pushd elaf
```

2- popd:

It is used to go back one step from where you are standing. It is used after the **pushd** command only, meaning that if you use the cd command to enter a folder and want to use the **popd** command to return one step, the command will not be executed and this command will be used alone, as follows:

```
elaf@the-game:/mnt/games/chm/elaf> popd
```

The result will be as follows:

```
/mnt/games/chm ~
elaf@linux-r58p:/mnt/games/chm>
```

Important note: The **popd** and **pushd** commands do not have manual help pages.

3- cd:

Used to move between folders and enter them.

1- To enter a folder in the same place where I am standing, type the following:

cd 🟥The folder Name

Example: To enter the folder **elaf** in the path I am currently standing in, type the following:

```
elaf@the-game:~> cd elaf
```

The result will be as follows:

```
elaf@the-game:~/elaf>
```

2- To enter a folder in a path other than the one you are standing on, type that:

cd █ path

Example: To enter the **ogg** folder in the following path **/ mnt / collection1**, type the following:

```
elaf@the-game:~> cd /mnt/collection1/ogg
```

The result will be as follows:

```
elaf@the-game:/mnt/collection1/ogg>
```

3- To go back one step, type the following:

```
elaf@the-game:/mnt/collection1/ogg> cd ..
```

The result will be as follows:

```
elaf@the-game:/mnt/collection1>
```

4- To return to the original place where you were standing, write the following:

```
 elaf@the-game:/mnt/collection1/ogg> cd ~
```

4- or

```
 elaf@the-game:/mnt/collection1/ogg> cd
```

or

```
 elaf@the-game:~> cd –
```

The result will be as follows:

```
 elaf@the-game:~>
```

In other words, [cd ~], [only cd] and [cd -] do the same function.

5- To move up twice, write the following:

```
elaf@the-game:/mnt/collection1/ogg> cd ../..
```

The result will be as follows:

```
elaf@the-game:/mnt>
```

4-pwd:

Used to display the path or the place where you are standing. And the way to use it is very simple, just write the bitter like this alone and only.

```
elaf@the-game:~> pwd
```

The result will be as follows:

```
/home/elaf
```

5-ls:

This command is used to display files and folders in the path you specify, or to display files and folders in the current path if you do not specify any paths for them.

1- To view all files and indexes in the current path {where you are currently standing} write the following:

```
elaf@the-game:~> ls
```

The result will be as follows:

```
bin Desktop Documents RealPlayer11GOLD.bin
public_html
```

Note that the result is arranged alphabetically.

2- To view all files and indexes in any path on the device, type the following:

```
elaf@the-game:~> ls /var
```

The result will be as follows:

```
adm cache lib log opt spool X11R6
agentx games lock mail run tmp yp
```

Note that the result is arranged alphabetically.

3- To view all hidden and non-hidden files and folders, type the following:

```
elaf@the-game:~> ls -a
```

The result will be as follows:

```
        .                   .gconf                        .qt
        ..                 .gconfd              .realplayerrc
```

```
   .bash_history           .gnome2                    .recently-used
       .bashrc                        .gnome2_private  .recently-
     used.xbel
       .beagle             .gnu-emacs                            .scim
          bin                                .gtk_qt_engine_rc .skel
       .config              .ICEauthority                   .subversion
  .DCOPserver_the-game__0 .inputrc          .thumbnails
     .DCOPserver_the-game_:0  .kde               .tvtime
           Desktop                .kermrc              .urlview
             .dmrc                         .local     .wapi
         Documents                .mailcap          .Xauthority
            .dvipsrc                     .mcop     .xchat2
             .emacs                     .mozilla          .xcoralrc
           .esd_auth                 .mplayer          .xemacs
              .exrc                     .muttrc        .xim.template
          .fontconfig               .ooo-2.0       .xine
              .fonts                     .opera       .xinitrc.template
           .fonts.conf                .profile       .xsession-errors
  RealPlayer11GOLD.bin              public_html                .xtalkrc
```

Note that the result is arranged alphabetically.

4- To view all files and folders without arrangement, use the U-Capital option as follows:

```
elaf@the-game:~> ls -U
```

The result will be as follows:

```
Documents Desktop bin public_html
RealPlayer11GOLD.bin
```

5- To view all files sorted by date instead of the alphabet, type the following:

```
elaf@the-game:~> ls -t
```

The result will be as follows:

```
Text File Desktop bin Documents public_html
RealPlayer11GOLD.bin
```

Here I created a new file called Text File to show the difference.

6- To view all files in an inverse way (arrange alphabetically from top to top) write the following:

```
elaf@the-game:~> ls -r
```

The result will be as follows:

```
Text File public_html RealPlayer11GOLD.bin Documents
Desktop bin
```

7- To view detailed information about files and folders such as (the type of file and the permissions granted to it, the number of hard links for the file, or the number of subfolders of the folder, the owner, the group, the size in bytes, the creation date, and finally the name of the file) type the following:

```
elaf@the-game:~> ls -l
```

The result will be as follows:

```
                  total 7384
  drwxr -xr-x 2 elaf users            4096 2008-06-12 15:01 bin
   drwx------ 10 elaf users     4096 2008-06- 18 13:47 Desktop
    drwx------ 2 elaf users                4096 2008-06-14 22:20
                 Documents
  drwxr -xr-x 2 elaf users                4096 2008-06-12 15:01
                 public_html
              -rwxrwxrwx 1 elaf users 7502048 2008-06-07 02:45
  RealPlayer11GOLD.bin          16 2008-06-13 11:25 Text File
```

```
-rw-r--r-- 1 elaf users
```

8- To view detailed information about files and folders as above, but with the size shown in kilograms, mega, and giga, write the following:

```
elaf@the-game:~> ls -hl
```

The result will be as follows:

```
 total 7.3M
drwxr-xr- x 2 elaf users 4.0K 2008-06-12 15:01 bin
drwx------ 10 elaf users 4.0K 2008-06-18 13:47 Desktop
drwx------ 2 elaf users 4.0K 2008-06-14 22:20 Documents
drwxr -xr-x 2 elaf users 4.0K 2008-06-12 15:01
public_html
-rwxrwxrwx 1 elaf users 7.2M 2008-06-07 02:45
RealPlayer11GOLD.bin
-rw-r--r-- 1 elaf users  16 2008-06-13 11:25 Text File
```

We note that at the beginning of the result, the total size of files and folders is written in mega unit, noting that the size of the folders does not include the size of the content of those folders.

9- To view all files and folders by size, type the following:

```
elaf@the-game:~> ls -S
```

For further clarification, we will combine option l with S, as follows:

```
elaf@the-game:~> ls -lS
```

The result will be as follows:

```
total 9760
-rwxrwxrwx 1 elaf users 7502048 2008-06-07 02:45
RealPlayer11GOLD.bin
-rw-r--r-- 1 elaf users 2111499 2008-06-18 04:23
Chikenux.01.7z
```

```
-rw-r--r-- 1 elaf users 284959 2008-06-18 04:07 83425-
opensusestreet.png
-rw-r--r-- 1 elaf users        11390 2008-06-18 02:30
hijra-0.1.2.tar.bz2
drwxr -xr-x 2 elaf users        4096 2008-06-12 15:01 bin
drwx------ 10 elaf users        4096 2008-06- 18 13:47 Desktop
drwx------ 4 elaf users        4096 2008-06-18 18:22
Documents
drwxr -xr-x 2 elaf users        4096 2008-06-12 15:01
public_html
-rw-r--r-- 1 elaf users        16 2008-06-13 11:25 Text File.txt
```

10- To view all files, folders and contents of folders also write the following:

```
elaf@the-game:~> ls -R
```

11- To view the files with their type, write the following:

```
elaf@the-game:~> ls -F
```

The result will be as follows:

```
bin/ Desktop/ Documents/ public_html/
RealPlayer11GOLD.bin*
```

We note that the folder takes the symbol / and the executable takes the symbol * and the link to another file or link is expressed with the symbol @ and socket takes the symbol =.

12- To view the file names separated by a **comma**, type the following:

```
elaf@the-game:~> ls -m
```

The result will be as follows:

```
bin, Desktop, Documents, public_html,
RealPlayer11GOLD.bin
```

13- To view the file names that start with a specific letter {such as a letter D}, type the following:

```
elaf@the-game:~> ls D*
```

14- To view all files and folders whose first letter is one of the following letters such as **z, R, r** and ends with a fixed section like **oot** type the following:

```
elaf@the-game:~> ls [Rrz]oot
```

The result will be as follows:

```
root:
Root:
```

We notice from this result that he found the root and Root folders which are empty and did not find a folder named zoot.

15- To view files and folders that start with the letter **a** through **r** and end with the section **llw** write the following:

```
elaf@the-game:~> ls [a-r]llw
```

The result will be as follows:

```
gllw:
kllw:
```

Note that it displays only the small letter from a to r and the letter will not display the capital.

16- To view files and folders in one column, type the following:

The result will be as follows:

```
agentx
cache
games
lib
lock
log
```

```
mail
opt
run
spool
tmp
X11R6
yp
```

17- To view the files and folders in alphabetical order, by extension, type the following:

```
elaf@the-game:~/Documents> ls -X
```

The result will be as follows:

```
    mido Chikenux.01.7z           hijra-0.1.2.tar.bz2 83425-
     opensusestreet.png
       tito RealPlayer11GOLD.bin 12.doc        Text File.txt
```

We note here that the folders come first because they have no extension and then the files come in alphabetical order of their extension as shown with a note that if the file has a compound extension such as (tar.bz2.) Only the last part of the extension is considered and here is (bz2.).

6-dir:

It is used to display files and folders just like the ls command and takes the same options as the ls command.

B- Working with files

1-touch:

- This command is used to create an empty file

- Used to change the date the file or folder was created, if it already exists.

1- To create an empty file without extension, for example, if you want to create the file with the name **asd**, type the following:

```
elaf@the-game:~> touch asd
```

2- To create an empty file with a specific extension {filename **xman** and its extension txt as}, write the following:

```
elaf@the-game:~> touch xman.txt
```

3- To create more than one file at a time, write the name of the two files one by one, leaving a space between them, write the following:

```
elaf@the- game:~> touch hima elaf
```

4- To change the date of the last amendment to a file or folder that already exists so that the date of the last amendment to the file is the moment of writing the command {and that if you have permission to write on that file or folder {write the same as the previous command as follows:

```
elaf@the- game:~> touch hima elaf
```

5- To change the date of the last modification of a file or folder that already exists with any other date chosen, the option t is used, but before explaining how to use this option you must explain how to write the date, as follows:

cc yy MM DD hh mm ss

cc ➔The first two digits of the four years.

yy ➔last two digits of the years.

An example to illustrate cc and yy:

The year 1985 is cc = 19 and yy = 85.

The year 2008 is cc = 20 and yy = 08.

If yy ranges from (0 - 68) then (cc = 20)

If yy is between (99 - 69) then (cc = 19) is considered.

If you do not write yy, the year is the current year.

MM ➔expresses the months.

DD ➔ express days.

mm ➔ expresses the minutes.

ss ➔expresses the seconds.

Important note:

For any number to be written with one digit in the history, for example: a nine day in the month to write 9, but to write 09.

If you want to place the seconds, you must precede it.

2-cat:

Uses of the cat command:

a- Read one file on the screen without being able to change that file.

b- Read more than one file together with merging them into one file on the screen only so that they appear under each other, according to the order of the files when writing the command.

c- Merging more than one file into one file and merging them under each of them according to the filename sequence when writing the command.

d- Writing inside a file and deleting the data that was in it.

e- For modification inside a written text file while preserving the data that was in it.

f- is used to create a new file and write it inside.

Important note:

it is not recommended to use bitter cat with long text files because it will be displayed on the screen more quickly than it is read.

The following is an explanation of these functions and uses:

1- To read one file on the screen without the possibility of modifying its content (assuming that the file is named **saeed**), write the following:

```
elaf@the-game:~> cat saeed
```

The result is to display the content of the file inside the terminal.

2- To view the file with the numbering of each line, type the following:

```
elaf@the-game:~> cat -n saeed
```

3- To read more than one file at once with merging it on the screen only (assuming that you have two files **saeed** and **tman**), write the following:

```
elaf@the-game:~> cat saeed tman
```

The result is the width of the two files, so that the **saeed** file is in the first, followed by the **tman** file.

4- To read more than one file at once with merging it on the screen only and numbering each line, write the following:

```
elaf@the-game:~> cat -n saeed tman
```

5- To merge more than one file into one new file (assuming that the new file is named mid) write the following:

```
elaf@the-game:~> cat saeed tman > mid
```

The result is to create a new file named mid containing the files inside the saeed and tman files.

6- To merge more than one file into one new file with the numbering of each line of the new file (assuming that the new file is named **dido**) write the following:

```
elaf@the-game:~> cat -n saeed tman > dido
```

The result is to create a new file named dido that contains the inside of the two files of data in addition to that the lines of this file are numbered.

7- To write inside a file and delete all the data that was inside it, write the following:

```
elaf@the-game:~> cat > saeed
```

After completing typing inside the file, press Enter and press (Ctrl + d) to save what you wrote.

8- To amend within a file written in it and add new data to it while preserving the data previously contained, write the following:

```
elaf@the-game:~> cat >> saeed
```

After completing typing inside the file, press Enter and press (Ctrl + d) to save what you wrote.

9- To create a new file with the script inside it (assuming that the new file is named **bibo**) write the following:

```
elaf@the-game:~> cat > bibo
```

After completing typing inside the file, press Enter and press (**Ctrl + d**) to save what you wrote.

3-tac:

The function of this command is to display the file from end to end. Any last line is the first line (assuming that the file to be displayed is named **bibo**) as follows:

```
elaf@the-game:~> tac bibo
```

4-file:

This time is used to display only the file type.

1- To display one file type (assuming that the file you want to display is called **bibo**) write the following:

```
elaf@the-game:~> file bibo
```

The result will be as follows:

```
bibo: ASCII text
```

2- To display more than one file type (assuming the files to be displayed are bibo, RealPlayer11GOLD.bin, and bin) write the following:

```
elaf@the-game:~> file RealPlayer11GOLD.bin bibo bin
```

The result will be as follows:

5-more:

Used to read files that are larger than a page. And the way to use it is very easy, which is to write the name of the file to be read after the command more, as follows (assuming that the file to be read is called tman):

```
elaf@the-game:~> more tman
```

In table 3.3 that show the keys used for command more:

Table 3.3: keys used for command more

Space	**Go to the next page.**
F	Go to the next page.
b	Return to the previous page.
Enter	Move to the next line.
h	for help.
q	Exit.

6-less:

It is used to read files just like the previous command more but with some improvements.

The way to use it just like the previous command more, as follows:

```
elaf@the-game:~> less tman
```

The same keys used with more are used with less as well as the possibility to use the up and down arrows for progress and lag line. For more information, write the following:

```
elaf@the-game:~> less –help
```

It is worth noting that the command less is the program that displays the manual pages inside the terminal.

7-head:

Used to display a specific number of lines from the beginning of the file.

Example: If you want to display the first 10 lines of a file named **tman**, type the following:

```
elaf@the-game:~> head -10 tman
```

The result, of course, is the display of the first 10 lines of that file.

8-tail:

Used to display a certain number of lines from the end of the file. It is also used to display a portion of the file while neglecting the rest.

1- If you want to display the last 15 lines of the file, write the following:

```
elaf@the-game:~> tail -15 tman
```

2- If you want to view the file while neglecting the first 16 lines of it, write the following:

```
elaf@the-game:~> tail +16 tman
```

The result is that the file is viewed from the beginning of line 16 to the end.

9-wc:

Used to find information about a specific file. Assuming that the file you want to inquire about is his name saeed.

1- To knowing the number of lines, write the following:

```
elaf@the-game:~> wc -l saeed
```

2- To find out the number of words, write the following:

```
elaf@the-game:~> wc -w saeed -
```

3- To find out the file size in bytes inside the file, write the following:

```
elaf@the-game:~> wc -c saeed
```

4- To find out the number of letters in the file, type the following:

```
elaf@the-game:~> wc -m saeed -
```

5- To know all the above information once, write the following:

```
  elaf@the-game:~> wc saeed
```

10-grep:

It is used to search for a word or a sentence inside the text files (with the phrase being placed in quotation marks).

Important note: you must consider the capital & small case in the word you are looking for.

1- If you want to search for a word, and let **linux** be inside a text file called **saeed**, write the following:

```
elaf@the-game:~> grep linux saeed
```

The result is the appearance of all lines containing that word.

2- If you want to search for a sentence and let **Linux kernel** be inside a text file named **saeed**, write the following:

```
elaf@the- game:~> grep "Linux kernel" saeed
```

The result is the appearance of all lines containing that sentence.

3- To display all the lines that contain a specific word and let linux be inside a text file called **saeed**, type the following:

```
elaf@the- game:~> grep -v linux saeed
```

The result is the display of all lines containing the word **linux**.

4- To view all the lines that contain a specific sentence and let **Linux kernel** be inside a text file named **saeed**, type the following:

```
elaf@the- game:~> grep -v "Linux kernel" saeed
```

The result is to display all lines containing the **Linux kernel** clause.

5- To view all the lines that contain a specific word and be Linux with the display of the numbers of that line, inside a text file called **saeed**, type the following:

```
elaf@the-game:~> grep -n linux saeed
```

6- To view all lines containing a specific sentence, and let **Linux Kernel** display the numbers of those lines within a text file named **saeed**, type the following:

```
 elaf@the- game:~> grep -n "Linux kernel" saeed
```

7- To view the number of lines containing a specific word, and to be **linux** in a text file called **soca**, type the following:

```
elaf@the- game:~> grep -c linux saeed
```

8- To view the number of lines containing a specific sentence, and let **Linux kernel** be in a text file named **saeed**, type the following:

```
elaf@the- game:~> grep -c "Linux kernel" saeed
```

9- To display file names only that contain a specific word, and let linux be in a specific path, then / home / elaf / * type the following:

```
elaf@the-game:~> grep -l linux /home/elaf/*
```

10- To display only file names that contain a specific sentence, and let Linux kernel be in a specific path, let it be / home / elaf / * Type the following:

```
elaf@the-game:~> grep -l "Linux kernel" /home/elaf/*
```

When searching for sentences and words using the grep command, it is preferable to use some symbols. These symbols are placed in the sentences and words that are the subject of research. These symbols in table 3.4 increase the efficiency of the search and are called those symbols (regular expressions pattern).

Table 3.4: symbol for increase the efficiency of the search

^	**Meaning that the word to search for is at the beginning of the line.**
$	Meaning that the word to search for is at the end of the line.
.	To make up for one minus character.
[First letter, second letter]	It revolves around words that begin with any letter Confined between these two letters.
Capital letter and lowercase letter Without spaces] followed by the rest of the word	Spin on the word that begins with the aforementioned letter either Akan Capital M small.

Examples of using regular expressions:

a- To search for lines that begin with the word **linux**, type the following:

```
elaf@the-game:~> grep ^linux saeed -
```

b- To search for lines that end with the word **linux**, type the following:

```
elaf@the-game:~> grep linux$ saeed
```

Important note:

If the line ends with a space or a period. The search will fail because the end of the line is not linux.

c- To search for words that start with the letter l and end with the letter x (and connect the word to turn on it is linux but I don't remember it exactly) write the following:

```
elaf@the-game:~> grep l...x saeed -d
```

d- To search for words beginning with the letters a through f, type the following:

```
elaf@the-game:~> grep [a,f] saeed-
```

e- To search for lines with the word root or Root type the following:

```
elaf@the-game:~> grep [Rr]oot saeed
```

11-zgrep:

It is used to search for a word or a sentence within the compressed text files (with the phrase enclosed in double quotes).

The **zgrep** command is the grep command, meaning we can apply all of the above with the grep command to the **zgrep** command. The only difference is that the files I search are compressed text files. Here's an example to explain:

If you want to search for a word and let **linux** be inside a zip text file called **saeed.tar.gz** write the following:

```
elaf@the-game:~> zgrep Linux saeed.tar.gz
```

12-diff:

It is used to compare two files line by line.

Example of clarification: If you have two files the first name is asd1 and contains the following line:

```
Elaf elaf
Ali elaf
```

And the second named asd2 contains the following lines:

```
Elaf elaf
Ali ali
```

To compare these two files, write the following:

```
elaf@the-game:~> diff asd1 asd2
```

Notes:

a- The two lines are considered different if the distances between the words are different even if the two lines carry the exact same words.

b- The two lines are considered different if one end has a distance while the other line does not end with a space.

c- The two lines are considered different if one end has a point while the other line does not end with a point.

C- Working with folders

1-mkdir:

It is used to create only one volume and also used to create several folders next to each other or within each.

1- To create an empty folder named **conan** in the current path, type the following:

```
elaf@the-game:~> mkdir conan
```

2- To create an empty folder named conan in the / mnt path, type the following:

```
the-game:/home/elaf # mkdir /mnt/conan
```

To create a file in this path you must be a root user.

3- To create several empty folders next to some (assuming the folders are amr, gigi, hani, and dido) write the following:

```
elaf@the-game:~> mkdir amr gigi hani dido
```

4- To create more than one folder inside some (assuming that the folders are edg, amr, gigi, hani and dido) write the following:

```
elaf@the-game:~> mkdir -p edg/amr/gigi/hani/dido
```

5- If you want the mkdir command to inform you about the progress of the creation process, i.e. informing you of the folders that it has finished creating, use the v- option as follows:

```
elaf@the-game:~> mkdir - vp edg/amr/gigi/hani/dido
```

2-tree:

It is used to display folders in tree form.

1- To display the folders and their contents of files and folders in the form of trees in the current path, write the following:

```
elaf@the-game:~> tree
```

2- To display the folders and their contents of files and folders in the form of trees in the path (/var), write the following:

```
elaf@the-game:~> tree /var
```

We notice that this method is impractical when the path that you display contains many files. The merit is to view folders only without displaying what you contain of files and folders.

3- To display only the folders and their contents in folders in the form of a tree in the current path, write the following:

```
elaf@the-game:~> tree -d
```

4- To display only the folders and their contents as folders in the form of a tree in the path (/ var) write the following:

```
elaf@the-game:~> tree -d /var
```

D- Copy, Move, Delete and Rename Commands

1-rm:

This command is used to delete files or folders, whether full or empty.

1- To delete a file and let it be called saeed present in the current path, write the following:

```
elaf@the-game:~> rm saeed
```

2- To delete a file named bibo located in / mnt / collection1 type the following:

```
elaf@the-game:~> rm /mnt/collection1/bibo
```

3- To forcefully delete the file and let it be named **koki**, write the following:

```
elaf@the-game:~> rm -f koki
```

4- To avoid accidentally deleting a file where it gives you a message whose answer is yes or no Assuming that the file is named nani and is in the current path, write the following:

```
elaf@the-game:~> rm -i nani
```

The result is as follows:

```
rm: remove regular file `nani'?
```

Type yes to delete:

```
rm: remove regular file `nani'? yes -
```

5- To delete all files in the current path, type the following:

```
elaf@the-game:~> rm *
```

6- To delete the volume, whether it is full or empty, type the following (assuming that its name is zizo):

```
elaf@the-game:~> rm -r zizo
```

Important Notes:

The rm command can delete files if these files have links or shortcuts. To delete these files, you must first delete these shortcuts. To see the number of shortcuts or links to the file, use the ls -l command.

We cannot delete files that we do not have permission to write on.

2-cp:

It is used to copy files and folders, and when using it, it must be specified where you will copy from and where you will be copied.

1- To copy a file named fofa from / mnt / collection1 to / home / elaf / Desktop, type the following:

```
elaf@the-game:~> cp /mnt/collection1/fofa
/home/elaf/Desktop
```

2- To copy a file named fofa from the current path to / mnt / collection1 type the following:

```
elaf@the-game:~> cp fofa /mnt/collection1
```

3- To forcibly copy a file named fofa from the current path to / mnt / collection1 type the following:

```
elaf@the-game:~> cp -f fofa /mnt/collection1
```

4- To avoid copying a file located in another place with the same name, use the -i option, and so it will ask you before copying if there is an existing file with the same name or not and the answer is yes or no, for example if you have a file named miro and want to copy it from the path / mnt / collection1 To the current path, using the -i option, where there is a file with the same name in the current path, type the following:

```
elaf@the-game:~> cp - i /mnt/collection1/miro ~
```

The result is as follows:

```
cp: overwrite `/home/elaf/miro'?
```

Type no until that file is copied.

```
cp: overwrite `/home/elaf/miro '? n
```

Note: The mark ~ indicates the current path.

5- To copy a folder, whether it is full or empty, use the -r or -R option. For example, if you have a folder named vector in the path / mnt / collection1 and want to copy it to the path **/ home / elaf / Desktop**, type the following:

```
elaf@the-game:~> cp -r /mnt/collection1/vector
/home/elaf/Desktop
```

Or

```
elaf@the-game:~> cp -R /mnt/collection1/vector
/home/elaf/Desktop
```

Note that when copying a folder or file, the owner changes to the user who copied the file or folder.

6- To copy a folder named vector from / mnt / collection1 to / home / elaf / Desktop but with the copying process showing EA to concluded it is copied and any to a copy that has not been copied (here we use the -v option) type the following:

```
elaf@the-game:~> cp -vr /mnt/collection1/vector /home/
elaf/Desktop
```

Question: Can a file or folder be copied to another place or in the same place with a different name?

Aljaba: Yes

7- If you want to copy a file named **fola** in the same place with a different name, and let it be soma, write the following:

```
elaf@the- game:~> cp fola soma
```

8- If you want to copy a file named fola from the current path to / mnt / collection1 with a different name and let it be soma write the following:

```
elaf@the-game:~> cp fola /mnt/collection1/soma
```

9- If you want to copy a file from one path to another so that I am not present in any of the two paths, for example, if you want to copy a file named elaf located in the path / mnt / collection1 to the path / mnt / collection2, type the following:

```
elaf@the-game:~>cp/mnt/collection1/elaf
/mnt/collection2
```

2-mv:

It is used to transfer files and folders from one place to another and when it is used it must be specified where you will be moved from and where you will be moved.

The **mv** command can be used to rename files and folders by moving the file or folder. To the same path with his name changed.

The use of bitter mv is very similar to the use of bitter cp except that it moves folders without using the -r option, here are examples of using the **mv** command below.

1- To transfer a file named **fofa** from / **mnt** / collection1 to / home / elaf / Desktop, type the following:

```
elaf@the-game:~> mv /mnt/collection1/fofa
/home/elaf/Desktop
```

2- To transfer a file named **fofa** from the current path to **/ mnt / collection1** type the following:

```
elaf@the-game:~> mv fofa /mnt/collection1
```

3- To transfer a file named fofa by force from the current path to / mnt / collection1 type the following:

```
elaf@the-game:~> mv -f fofa /mnt/collection1
```

4- To avoid transferring a file to another place with the same name, use the **-i** option, and so it will ask you before the transfer if there is an existing file with the same name or not and the answer is yes or no. For example, if you have a file named **miro** and want to transfer it from the path **/ mnt / collection1** to the path Current With the use of the **-i** option where there is a file with the same name in the current path, type the following:

```
elaf@the-game:~> mv -i /mnt/collection1/miro ~
```

The result is as follows:

```
mv: overwrite `/home/elaf/miro'?
```

Type **no** so that the file does not transfer.

Note: The mark ~ indicates the current path.

5- To move a folder full of files and subfolders and let it be named **zima** from the current path to the path **/ mnt / collection1**, type the following:

```
elaf@the-game:~> mv zima /mnt/collection1
```

6- To transfer a folder named **zima** from **/ mnt / collection1** to **/ home /elaf / Desktop** but with the transfer shown (here we use the -v option) type the following:

```
elaf@the-game:~> mv - v /mnt/collection1/zima
/home/elaf/Desktop
```

7- To rename a file or folder, do the following:

The file or folder is moved to the same location but with a different name, for example if you have a folder named zima found in **/ home / elaf / Desktop** and you want to rename it. to mike, write the following:

```
elaf@the-game:~> mv /home/elaf/Desktop/zima
/home/elaf/Desktop/mike
```

Note:

```
elaf@the-game:~> cd /home/elaf/Desktop
elaf@the-game:~/Desktop> mv zima mike
```

If you stand on the path where the folder is to be renamed, you will not need to write all of these paths. Here is an example to clarify the same previous example:

```
elaf@the-game:~> cd /home/elaf/Desktop
elaf@the-game:~/Desktop> mv zima mike
```

When changing the name of a file or folder using the mv command and there was in that path a folder with the same new name you will transfer the file or folder to which you want to change its name because the mv command first searches for a folder with the new name in order to move the file or folder to it.

But if you want to change the name of a file and there is in that path a file with the same new name, overwrite will be done, as the data in the file to be renamed

biso wanted to change his name to nona and by chance there was a file named nona that you did not notice. How is the result?

The result is that when writing the command.

```
elaf@the-game:~/Desktop> mv biso nona
```

All data in the bios file goes to replace the data in the **nona** file.

8- If you want to transfer a file named **fola** from the current path to **/ mnt / collection1** with a different name and let it be soma write the following:

```
elaf@the-game:~> mv fola /mnt/collection1/soma
```

9- To make a backup copy of the file to be moved or renamed, use the **-b** option, assuming that the file is named **wama** as follows:

```
elaf@the-game:~> mv -b wama /mnt/collection1
```

The result is to make a backup copy in the place of the original file before transferring it under the name wama **~** with moving the wama file to the path **/ mnt / collection1**.

Copy and move with a tag *:

To copy all files and folders in the current path once to a specific path, then / mnt / collection1 type the following:

```
elaf@the-game:~> cp -r * /mnt/collection1
```

To transfer all files and folders in the current path once to a specific path, then / mnt / collection1 type the following:

```
elaf@the-game:~> mv * /mnt/collection1
```

important note:

When writing the name of a folder or file consisting of two words, including a space in the terminal, the name of the folder or file must be placed between the quotation marks “For example, type “new folder” and do not write a new folder.

3-ln:

Used to create soft and hard bonds.

1- Soft links:

A **soft link** is a small file that contains the path of the original file or folder. If you delete this link, this will not affect the original file. But if you delete the original file, the link will not work. Soft links to files and folders are made equally.

Soft links work through:

{Different file systems - network file systems - installed devices – function}.

To create soft links, we use the ln command with the -s option. For example, if you have a file named **hima** that is in **/ mnt / collection1** and you want to create a link with it named **zima** in **/ home / elaf / Desktop**, type the following:

```
elaf@the-game:~> ln -s /mnt/collection1/hima
/home/elaf/Desktop/zima
```

2- Solid links:

To enable the creation of hard links via: **{network file systems - different file systems - partition devices}** and also it is not possible to create a solid link to a folder. That is, the hard joint must be in the same partition as the original file.

To create a hard link, use the command ln with no option. For example, if you have a file named nova in the path **/ mnt / collection1** and want to create a hard link with the name **koka** in the path **/ mnt / collection1** / ogg, type the following:

```
elaf@the-game:~> ln /mnt/collection1/nova
/mnt/collection1/ogg/koka
```

Compression and archiving

A- Archiving:

An archive is a collection of files and folders located together in one place and treated as a single file.

1-tar:

Used to create archives, to view archive contents, and to decrypt archives. The archived files and folders are under the. Tar extension.

a- Archiving work

To create an archive, use the **-cvf** option, where c means creat, and v means display operations during the moderation process, and f means file. It is worth noting that this option is not interchangeable, that is **(-cvf** is not equal to **-fvc**).

1- To create an archive of a folder named **wcw** with a set of files and subfolders located in the path **/ mnt / collection1** with the archive placed in the path **/ home / elaf / Desktop**, write the following:

```
elaf@the-game:~> cd /mnt/collection1
elaf@the-game:/mnt/collection1> tar -cvf /home/ahmed/
Desktop/wcw.tar wcw
```

2- To create an archive of a folder called **wcw** with a set of files and subfolders it is in the path **/ mnt / collection1** and you want to place the archive in the same path as the **wcw** folder and name it **mtm**, type the following:

```
elaf@the-game:~> cd /mnt/collection1
elaf@the-game:/mnt/collection1> tar -cvf mtm.tar wcw
```

3- If you want to create an archive for a folder with a set of files and subfolders and you want to select some of these files and folders to be aware within the archive so that it gives you a message asking you about each file or folder do you want to include it for the archive or not and the front y or n use the -cwf option.

Assuming that the folder you want to archive is named **gigi** in path **/ mnt / collection1** and you want to put the archive in the same path as the **gigi** folder and name it **vino**, write the following:

```
elaf@the-game:~> cd /mnt/collection1
elaf@the-game:/mnt/collection1> tar -cwf vino.tar gigi
```

The result is as follows:

```
add `/mnt/collection1/gigi'?
```

You must type y to complete the archiving process.

```
add `/mnt/collection1/gigi'?y
```

Thus, he will follow up with you about the questions for each file or subfolder.

We note that if you answer the first question n it will not ask you about the rest of the files and subfolder will then receive an empty archive.

4- To archive a folder named Raya using the formula **gzip** with the archive placed in the same folder path and name it hoda.tgz write the following:

```
elaf@the-game:~> tar -czvf hoda.tgz raya
```

5- To archive a folder named raya using the formula bzip2 with the archive placed in the same folder path and name it hoda.bz2 type the following:

```
elaf@the-game:~> tar -cjvf hoda.bz2 raya
```

b- View the archive contents:

1- To view the contents of an archive named vino.tar found in / mnt / collection1 type the following:

```
elaf@the-game:~> tar -tf /mnt/collection1/vino.tar
```

2- To view the contents of an archive named vino .tar found in / mnt / collection1 with all data for files and subfolders in it write the following:

```
elaf@the-game:~> tar -tvf /mnt/collection1/vino.tar
```

c- Decompile an archive and extract its contents:

1- To extract the contents of an archive named vino.tar found in / mnt / collection1 so that the extracted files are placed in the same archive path, type the following:

```
elaf@the-game:~> cd /mnt/collection1
elaf@the-game:/mnt/collection1> tar -xvf vino.tar
```

2- To extract the contents of an archive named vino.tar found in **/ mnt / collection1** so that the extracted files are placed in the current path, type the following:

```
elaf@the-game:~> tar -xvf /mnt/collection1/vino.tar
```

3- To extract one or a few files from an archive named vino.tar found in / mnt / collection1, type the following:

```
elaf@the-game:~> tar -xvwf /mnt/collection1/vino.tar
```

The result is as follows:

```
extract `gigi'?
```

You must write y to complete the decomplication process.

```
extract `gigi'?y
```

Thus, he will follow up with you about the questions about each file or subfolder.

4- If you want to extract a specific file named nana.pdf from an archive named wcw.tar found in **/ mnt / collection1** type the following:

```
elaf@the-game:~> cd /mnt/collection1
elaf@the-game:/mnt/collection1> tar -xf wcw.tar
wcw/nana.pdf
```

d- Add a file or folder to an archive:

1- To add a file named **mido** to an existing archive named yara.tar knowing that the file and the archive are in the same path, write the following:

```
elaf@the-game:~> tar -rvf yara.tar mido
```

2- Adding a folder (containing a set of sub files) named gamal to an existing archive whose name is yara.tar. Knowing that the file and the archive are in the same path, write the following:

```
elaf@the-game:~> tar -rvf yara.tar gamal
```

2-cpio:

Used to create an archive, to extract files from the archive, and to insert files into an archive.

When creating a cpio archive, a list of file and folder names to be archived is then sent to the standard output or to the device that will be archived by the -f option. Usually the ls or find command is used to provide that list to the standard output.

1- To archive a folder named noga {with only a set of files} found in / mnt / collection1 with the archive placed in the same path as the noga folder and name it tatu, type the following:

```
elaf@the-game:~> cd /mnt/collection1/noga
elaf@the-game:/mnt/collection1/noga> ls | cpio -ov >../tatu.cpio
or
ahmed@the-game:/mnt/collection1/noga> find | cpio -ov >
../tatu.cpio
```

2- To archive a folder named noga {with a set of files and subfolders} found in **/ mnt / collection1** with the archive placed in the same path as the **noga** folder and name it **tatu**, type the following:

```
elaf@the-game:~> cd /mnt/collection1/noga
elaf@the-game:/mnt/collection1/noga> find | cpio -ov >
../tatu.cpio
```

Option **o** means create. The option v is used to show what is finished archiving.

B- Compression:

1-gzip:

It is used to compress files as it is used to compress any tar type archive. For example, the **tar.gz** extension is a tar archive compressed by the **gzip** command.

Notes:

a- The volume is not compressed directly, rather it must be first archived and after that it is compressed.

b- After compressing the file or archive, it will be erased or removed.

Before explaining the matter, you must know that the pressure levels range from (9-1) and the pressure level can be determined or left default.

1- To compress a file named **amr -diab.mp3** with the **zip** file placed in the same path to the file and name it with the same name and leave the compression level default, type the following:

```
elaf@the-game:~> gzip amr-diab.mp3
```

2- To compress an archive named amr-diab with the archive placed in the same path as the archive and name it with the same name and choose the maximum pressure level (9), write the following:

```
elaf@the-game:~> gzip -9 amr-diab.tar
```

3- To make sure that the compressed file is in proper condition or not, use the -tv option, for example if you have a compressed archive named **amr- diab.tar.gz** and want to make sure that the compressed file is in proper condition, type the following:

```
elaf@the-game:~> gzip -tv amr-diab.tar
```

The result is either (**amr-diab.tar.gz: ok**): the file is valid.

Or (**amr-diab.tar.gz: not ok**) the file is corrupt.

4- To view all information about a compressed file named **amr-diab.tar.gz**, such as the file size before and after compression, and the compression ratio, write the following:

```
elaf@the-game:~> gzip -l amr-diab.tar.gz
```

5- To unzip a zip file named amr-diab.mp3.gz in the same path, type the following:

```
elaf@the-game:~> gzip -d amr-diab.mp3.gz
```

Note:

When the compressed file is decomposed, it is automatically deleted and the unzipped file remains.

Reference

1- Marsh, Matthew G. Policy routing using Linux. Sams, 2001.

2- Guttman, Joshua D., et al. "Verifying information flow goals in security-enhanced Linux." Journal of Computer Security 13.1 (2005): 115-134.

3- Kofler, Michael. Linux: Installation, Configuration, Use. Addison-Wesley Longman Publishing Co., Inc., 1997.

4- Dague, Sean. "System installation suite massive installation for linux." Ottawa Linux Symposium. Vol. 93. 2002.

5- Morris, James, Stephen Smalley, and Greg Kroah-Hartman. "Linux security modules: General security support for the linux kernel." USENIX Security Symposium. ACM Berkeley, CA, 2002.

6- Wright, Chris, et al. "Linux security module framework." Ottawa Linux Symposium. Vol. 8032. 2002.

7- Zhang, Wensong. "Linux virtual server for scalable network services." Ottawa Linux Symposium. Vol. 2000. 2000.

8- Flickenger, Rob. Linux server hacks. " O'Reilly Media, Inc.", 2003.

9- Benvenuti, Christian. Understanding Linux network internals. " O'Reilly Media, Inc.", 2006.

10- McQuillan, Jim. "The Linux Terminal Server Project: Thin Clients and Linux." Annual Linux Showcase & Conference. 2000.