



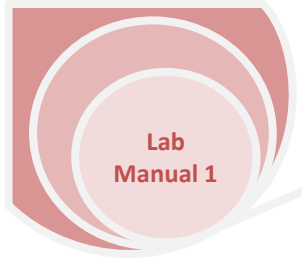
CS480-102

# Introduction to Linux

## Operating Systems

Introduction to Linux & Basic Commands

Fozia Noor  
Lab Manual 1



## Contents

<b>Lab Objectives</b> .....	3
<b>PART 1: Introduction to Linux</b> .....	4
What is Linux.....	4
What is Kernel.....	5
What's Linux Shell? .....	5
File System Organization.....	6
<b>PART 2: Linux Quick Start</b> .....	7
Graphical mode.....	7
Ubuntu introduction .....	7
Genome Desktop .....	8
Virtual Desktops.....	10
<b>PART 3: Linux basics Commands</b> .....	12
Navigation Commands.....	12
Commands Structure .....	15
Man command.....	15

CS480-102

## Lab Objectives

---

The primary objectives of this lab session are as follows

- Introduction to Linux
- Introduction to Ubuntu Environment
- Understanding Linux File System, Kernel & Shell
- Getting familiar with Linux Commands
  - Navigation commands
  - Commands Structure
  - Help Commands
  - Some Basic commands

For help regarding learning Linux environment, below is a short list containing some great examples; these sites have a lot of screenshots that will give you a glimpse of what Linux on the desktop can be like:

- <http://www.gnome.org>
- <http://kde.org/screenshots/>
- <http://www.openoffice.org>
- <http://www.mozilla.org>

## PART 1: Introduction to Linux

---

### What is Linux

Linux is an implementation of UNIX. The Linux operating system is written in the C programming language. Linux uses GNU tools, a set of freely available standard tools for handling the operating system.

Linux is a powerful & very popular operating system among the scientific, engineering, and academic communities due to its multi-user and multi-tasking environment, flexibility and portability, electronic mail and networking capabilities, and the numerous programming, text processing and scientific utilities availability. The Linux system is mainly composed of three different parts: the **kernel**, the **file system**, and the **shell**.

1. The **kernel** is that part of the system which manages the resources of whatever computer system it lives on, to keep track of the disks, tapes, printers, terminals, communication lines and any other devices.
2. The **file system** is the organizing structure for data. The file system is perhaps the most important part of the UNIX operating system. The file system goes beyond being a simple repository for data, and provides the means of organizing the layout of the data storage in complex ways.
3. The **shell** is the command interpreter. Although the shell is just a utility program, and is not properly a part of the system, it is the part that the user sees. The shell listens to your terminal and translates your requests into actions on the part of the kernel and the many utility programs

## What is Kernel

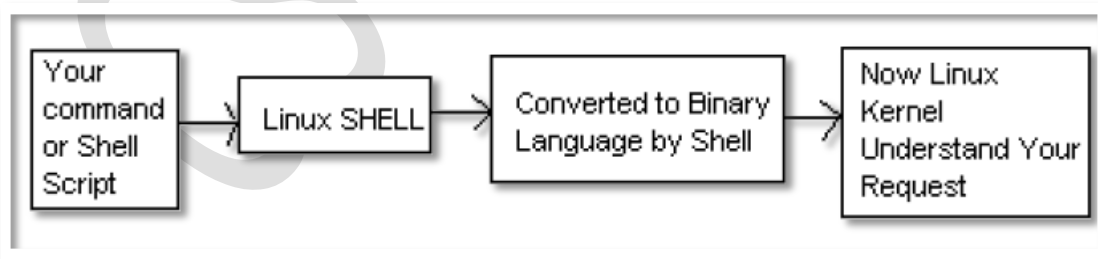
Kernel is part of Linux O/S. It manages resource of Linux O/S. Resources means facilities available in Linux. For example Facility to store data, print data on printer, memory, file management etc. Kernel decides who will use this resource, for how long and when. It runs your programs (or set up to execute binary files) its Memory resident portion of Linux. Its performance following task:-

1. I/O management
2. Process management
3. Device management
4. File management
5. Memory management

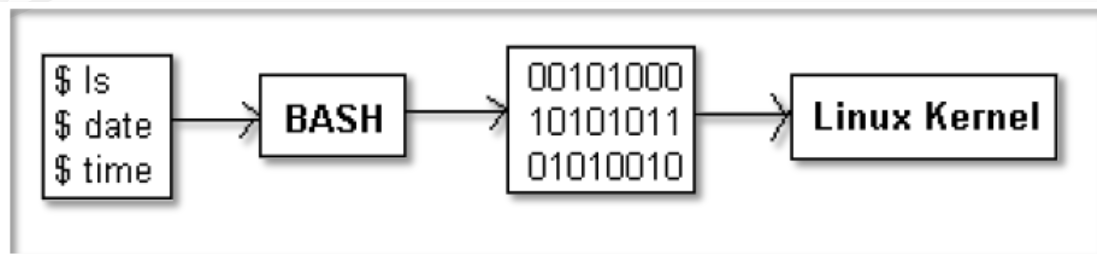
## What's Linux Shell?

Computer understand the language of 0's and 1's called binary language, In early days of computing, instruction are provided using binary language, which is difficult for all of us, to read and write. So in OS there is special program called Shell. Shell accepts your instruction or commands in English and translates it into computers native binary language

This is what Shell Does for US



You can type your command and shell converts it as



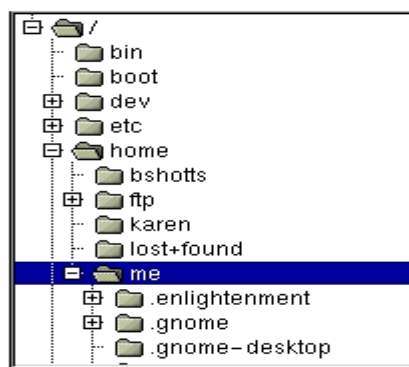
**Its environment is provided for user interaction.** Shell is a command language interpreter that executes commands read from the standard input device (keyboard) or from a file. Linux may use one of the following most popular shells.

- BASH ( Bourne-Again SHell )
- CSH (C SHell)
- KSH (Korn SHell)

Any of the above shell reads command from user (via Keyboard or Mouse) and tells Linux O/s what users want. If we are giving commands from keyboard it is called command line interface ( Usually in-front of \$ prompt, This prompt is depend upon your shell and Environment that you set or by your System Administrator , therefore you may get different prompt ).

## File System Organization

The files on a Linux system are arranged in what is called a hierarchical directory structure. This means that they are organized in a tree-like pattern of directories (called folders in other systems), which may contain files and other directories. The first directory in the file system is called the root directory



which is represented by a / (slash).. The root directory contains files and subdirectories, which contain more files and subdirectories and so on and so on.

In Linux, everything is treated as a file. A directory is a file. It is a file that contains a list of files and information belonging to those files. This would include things like who “owns” (created) the file, how long it is, and who can use it. Since a directory is simply a list of files, it can contain any file in it, including other directories.

## PART 2: Linux Quick Start

---

Linux systems have two basic modes for a system to run in: either quick and sober in **text console mode**, which looks like DOS with mouse, multitasking and multi-user features, or in **graphical mode**, which looks better but takes more system resources.

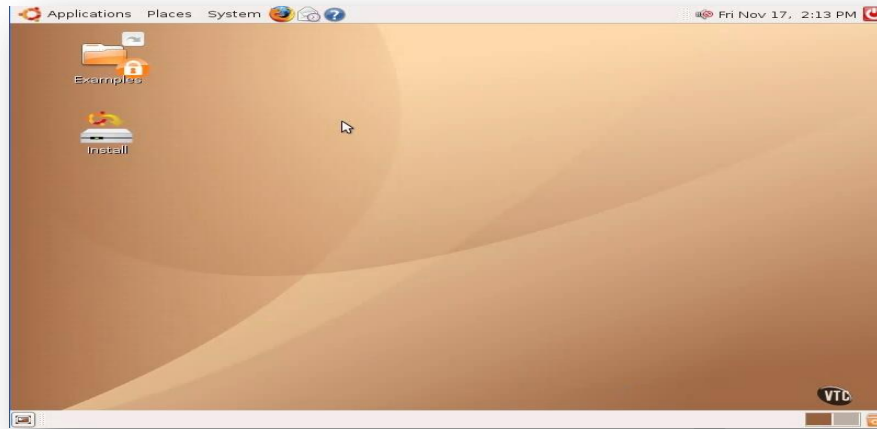
### Graphical mode

Below is given brief introduction to the environment of Ubuntu Operating System

### Ubuntu introduction



## Genome Desktop

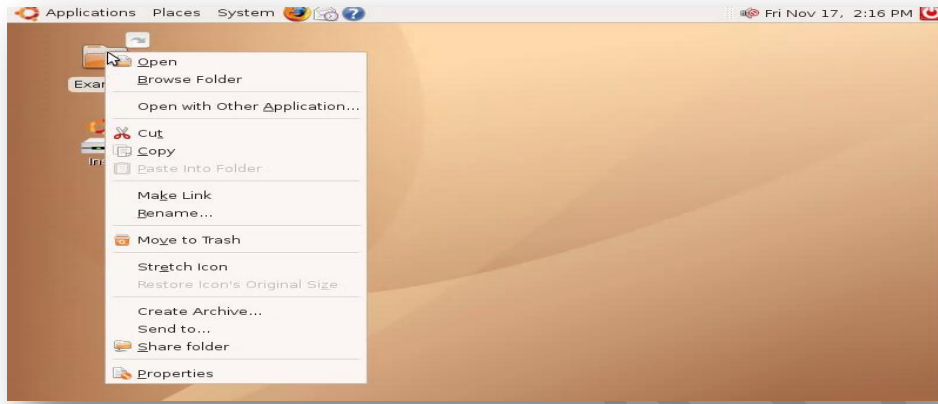


This Desktop environment is gnome. The other most famous desktop environment is KDE. **Application** menu in Gnome is same as start up menu in windows all installed application is here. **Places** give an easy access to other location and folders. It is just a short cut to places. **System** contains the log of very help full tools and utilities.

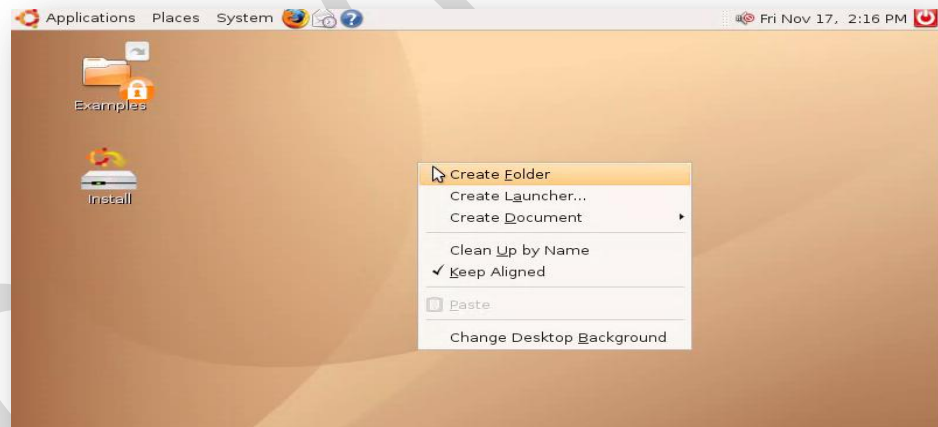




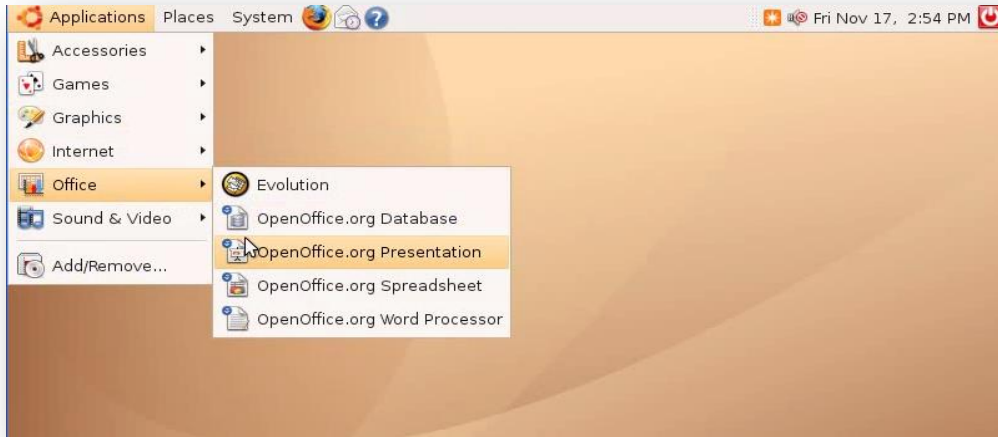
Like windows the Right clicks option is also available in Linux.



If you right click on the desktop then the menu changes



In applications Office application is installed which is just like a Microsoft documents and can open Microsoft Documents.



## Virtual Desktops

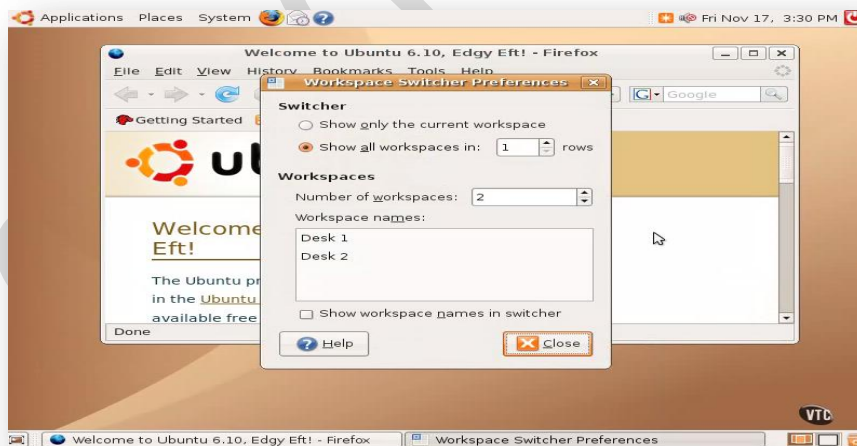
The option that Switch between work spaces is given at the right corner of Desktop. The idea is to provide more than one desktops. Suppose in your first workspace you are running any game application like



Switch to other desktop and open web browser.



The number of workspaces can be changed by right clicking on the workspace area. Now click on preferences this window will be popped up and you can change the number of work spaces from there.



You can customize the desktop can change themes, can add new elements etc

## PART 3: Linux basics Commands

---

### Navigation Commands

#### pwd

Since a command line interface cannot provide graphic pictures of the file system structure, it must have a different way of representing it.

The directory you are standing in is called the **working directory**. To find the name of the working directory, use the `pwd` command.

```
[root@localhost ~]# pwd
/root
```

When you first log on to a Linux system, the working directory is set to your home directory. This is where you put your files. On most systems, your home directory will be called **/home/your\_user\_name**,

#### ls

To list the files in the working directory, use the `ls` command.

```
[root@localhost ~]# ls
anaconda-ks.cfg  Download      install.log.syslog  Public
Desktop         f            Music              Templates
Documents       install.log  Pictures           Videos
```

Create a new text file and save it in your home directory and again enter command `ls` to check that this time it shows your newly created file.

## Cd

To change your working directory (where you are standing right now) you use the `cd` command. To do this, type `cd` followed by the pathname of the desired working directory. A pathname is the route you take along the branches of the tree to get to the directory you want.

There is a directory called "Documents" inside root directory if u wants to go inside this directory the command will be

```
@localhost ~]# cd Documents
@localhost Documents]#
```

If u want to see the content of this directory type `ls`

```
[root@localhost Documents]# ls
FoziaDoc HadiDoc
```

if u want to see the path of Documents Directory from root

```
[root@localhost Documents]# pwd
/root/Document
```

Now let's say that we wanted to change the working `/root/Documents/HadiDoc`

```
[root@localhost ~]# cd /root/Documents/HadiDoc
[root@localhost HadiDoc]# pwd
/root/Documents/HadiDoc
```

You can specify a file or directory by its path name. There are two ways of expressing the path name: Full (absolute) path name or relative path name. The full path name starts with the root. `/`, and follows the branches of the file system, each separated by `/`, until you reach the desired file.

However, a relative path name specifies the path relative to your current working directory. Relative path name are more convenient because they are shorter

First, with an absolute pathname:

```
[root@localhost HadiDoc]# cd /root/Documents
[root@localhost Documents]#
```

Or relative path

If u wants to come back to the root directory then again use cd command

```
[root@localhost HadiDoc]# cd ..
[root@localhost Documents]#
```

we have to introduce two special directory entries:

- . the current directory
- .. the parent directory

A file named "salam.txt" is placed inside the directory "st111111".

The absolute (full) name is:

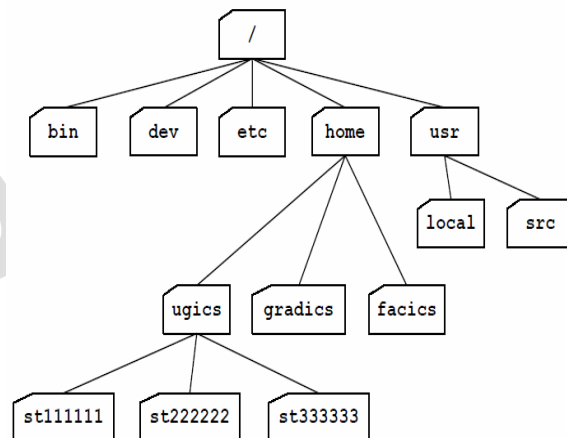
**/home/ugics/st111111/salam.txt**

The relative name is:

**st111111/salam.txt**

OR

**./st111111/salam.txt**



### Example

A directory named "ahmed" is placed inside the directory "gradics".

The absolute (full) name is: **/home/gradics/ahmed**

The relative name is: **../gradics/ahmed**

### Example

A file named "csh" is placed inside the directory "bin".

The absolute (full) name is: **/bin/csh**

The relative name is: **../bin/csh**

## Shortcut

If you type `cd` followed by nothing, `cd` will change the working directory to your home directory.

```
[root@localhost HadiDoc]# cd  
[root@localhost]#
```

## Commands Structure

In Linux most commands operate like this:

**command -options arguments**

where *command* is the name of the command, *-options* is one or more adjustments to the command's behavior, and *arguments* is one or more "things" upon which the command operates.

First of all we will Discuss Help Commands

## HELP Commands

If you don't know how to do something, Linux has a built-in help system that you can use. The most important tools are listed below

## Man command

To understand the working of the command and possible options use an command

```
[root@localhost ~]# man  
What manual page do you want?  
[root@localhost ~]# man ls
```

it will display the complete manual page. To exit the man utility and return to your prompt type "q" at any time

## info, info command

Using the GNU Info System

## whatis command

It List Description of a Program

```
[root@localhost ~]# whatis ls
ls          (1)  - list directory contents
ls          (1p) - list directory contents
[root@localhost ~]#
```

## '--help'

It outputs usage information about the tool, including the options and arguments the tool takes.

```
[root@localhost ~]# passwd --help
Usage: passwd [OPTION...] <accountName>
-k, --keep-tokens      keep non-expired authentication tokens
-d, --delete           delete the password for the named account
(-l, --lock           lock the named account (root only)
And so on .....
```

To find out the options that can be used with the “passwd” command type

```
[root@localhost ~]# passwd --help
```

## which command

It shows the full path to command

```
[root@localhost ~]# which ls
/bin/ls
[root@localhost ~]# which ps
/bin/ps
[root@localhost ~]#
```



## ? command

To see the switches that can be used with the reboot command type

```
[root@localhost ~]# reboot ?
usage: reboot [-n] [-w] [-d] [-f] [-h] [-i]
-n: don't sync before halting the system
-w: only write a wtmp reboot record and exit
and so on .....
```

## Some Basic Commands

### clear

Type **clear** to clean up your command prompt window.

This is especially helpful when you are typing lots of commands and need a clean window to help you focus. This is also useful when you are getting ready to type a rather long command and do not wish to become confused by other details on the screen.

```
[root@localhost ~]# clear
```

### Using Auto completion

Sometimes only the first letter or first two letters of a command are known. You can use the auto completion feature of the shell to list all the utilities/commands that match the first few letters of the command that you type by using the <TAB> key.

Assume you want to reboot the system for instance, but you can only remember that the command to do this starts with the letter "r"

Type "r" and then hit the [TAB] key twice. If prompted to display all possibilities type "y" (yes)

```
[root@localhost ~]# r[TAB][TAB]
Display all 999 possibilities? (y or n)y
racoons      read         resize       rmdir
rstartd
.....
```

Use the space bar to scroll through the list of commands that start with the letter “r”.

## whoami

it is used to displays the login name of the current user, The computer should respond with your login name.

```
[root@localhost ~]# whoami  
root
```

## finger

Use **finger** to see who's on the system

Typing **finger** allows you to see who else is on the system or get detailed information about a person who has access to the system.

```
[root@localhost ~]# finger  
Login      Name      Tty      Idle  Login Time  Office  
Office Phone  
root       root      pts/0    Apr 18 09:00 (:0.0)  
[root@localhost ~]#
```

Type **finger** followed by the name of a user's account to get information about that user.

```
[root@localhost ~]# finger root  
Login: root                               Name: root  
Directory: /root                          Shell: /bin/bash  
On since Sat Apr 18 09:00 (PKT) on pts/0 from :0.0  
1 second idle  
Mail last read Wed May 6 08:47 2009 (PKT)  
No Plan.  
[root@localhost ~]#
```

## echo

it tells the computer to retype the string after echo.

```
[root@localhost ~]# echo hello  
hello
```

## Date

Use **date** to set or check your server's date and time

```
[root@localhost ~]# date
Sat Apr 18 08:55:37 PKT 2009
```

## df

Use **df** to check disk space

Typing **df** provides a very quick check of your file system disk space.

```
[root@localhost ~]# df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/sda6              29753556    4428044  23789700  16% /
tmpfs                  1002980         0    1002980   0% /dev/shm
/dev/sdc1              3903368     297240   3606128   8%
/media/FOZIANOOR
```

```
[root@localhost ~]#
```

## Logout

logout is used to quit using the system. logout will log your account out of the system. Type logout at the prompt to disconnect from your Linux machine or to logout a particular user session from the system.

## ShutDown

The system is intended to be shutdown by the system administrator using the shutdown command in one of the forms shown below. Many systems are set up to capture the <CTRL><ALT><DEL> keystroke combination to issue the shutdown command through the init program. This will work on most systems if the root user is logged in. Examples of using the shutdown command are shown below

```
shutdown -h now
shutdown -r +10 "Rebooting in 10 minutes"
shutdown -r 13:00
```

The first command will shut down and halt the system immediately. The second will reboot the system in 10 minutes and send the message to all users. The third command will shut the system down and do a reboot at 1:00 in the afternoon

### **shutdown command**

To find out the arguments/options that can be used with the shutdown command type

```
[root@localhost ~]# shutdown -h
Usage:      shutdown [-akrhHPfnc] [-t secs] time [warning
message]
-a:         use /etc/shutdown.allow
-k:         don't really shutdown, only warn
-r:         reboot after shutdown
And so on.....
```