ଓଡ଼ିଶା ରାଜ୍ୟ ମୁକ୍ତ ବିଶ୍ୱବିଦ୍ୟାଳୟ, ସମ୍ବଲପୁର, ଓଡ଼ିଶା

Odisha State Open University, Sambalpur, Odisha

Established by an Act of Government of Odisha.

DIPLOMA IN CYBER SECURITY

DCS-01

Operating System Basics

Block

2

Unit -3

Linux Part-I

Unit -4

Linux Part-II

ଓଡ଼ିଶା ରାଜ୍ୟ ମୁକ୍ତ ବିଶ୍ୱବିଦ୍ୟାଳୟ, ସମ୍ବଲପୁର, ଓଡ଼ିଶା
Odisha State Open University, Sambalpur, Odisha
Established by an Act of Government of Odisha.

EXPERT COMMITTEE

Dr.P.K.Behera (Chairman)
Reader in Computer Science
Utkal University
Dr J.R.Mohanty (Member)
Prof And HOD
KIIT University
Dr.R.N.Behera (Member)
Senior Technical Director
NIC, ,Bhubaneswar
Dr.R.N.Behera (Member)
Senior Technical Director
NIC, ,Bhubaneswar
Sh Malaya Kumar Das (Member)
Scientist –E
NIC, Bhubaneswar
Dr.Bhagirathi Nayak (Member)
Professor And Head(IT & System)
Sri Sri University
Dr.Manoranjan Pradhan (Member)
Professor And Head
G.I.T.A
Mr V.S.Sandilya (Convener)
Consultant I.T
Odisha State Open University, Sambalpur,

DIPLOMA IN CYBER SECURITY

Course Writers

V.S.Sandilya
Academic Consultant (IT)
Odisha State Open University, Odisha

Mr Dhruba Charan Pradhan
.Lecturer
Department Of Computer Science
JKBK Government College ,Cuttack Odisha

Course Editor

Amaresh Sahu
Associate Professor
ABIT, Cuttack, Odisha

Unit -3
Linux Part-I

Learning objectives

After the Completion of this unit you should be able to know

1) About Linux
2) Linux vs Windows
3) Why use Linux
4) History of Linux
5) Different types of Linux environment.
6) Linux Commands.
7) File & Directory Management Commands
8) Know Kernel, Gnome of Linux

Structure

3.1 Introduction

Linux is a UNIX-based operating system. Its original creator was a Finnish student name ***Linus Torvalds,*** although being 'open source' it has change a great deal since its original conception. It belongs to nobody, and is free to download and use. Any changes to it are open for all to adopt, and as a result it has developed into a very powerful OS that is rapidly gaining in popularity worldwide, particularly among those seeking an alternative to Windows.

What is Linux?

Linux is an Operating System (OS). An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply – the operating system manages the communication between your software and your hardware. Without the operating system (often referred to as the "OS"), the software wouldn't function.

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

Linux is a complete multitasking, multiuser operating system, as are all other versions of UNIX. This means that many users can log into and run programs on the same machine simultaneously.

Linux is a free Unix-type Operating System (OS) for computer devices. The Operating System is what makes the hardware work together with the software. The OS is the interface that allows you to do the things, which you want with your computer. Linux is freely available to everyone. Linux gives you a graphical interface that makes it easy to use your computer, yet it still allows those with know-how to change settings by adjusting 0 to 1. It is only the kernel that is named Linux, the rest of the Operating System are GNU tools.

Linux vs. Windows

The main benefits and advantages of Linux over other operating systems, particularly Microsoft Windows, are:

- It is free to use and distribute.
- Support is free through online help sites, blogs and forums.
- It is very reliable – more so than most other operating systems with very few crashes.
- A huge amount of free open source software has been developed for it.
- It is very resistant to malware such as spyware, adware and viruses.
- It runs in a wide variety of machines than cannot be updated to use newer Windows versions.
- Since the source code is visible, 'backdoors' are easily spotted, so Linux offers greater security for sensitive applications.
- Linux offers a high degree of flexibility of configuration, and significant customization is possible without modifying the source code.

The Linux operating system is widely used by both home and business users, and its usage is increasing daily. It is considered that Linux will eventually overtake Microsoft Windows as the most popular operating system, which could also open the door further for more free software such as Open Office, The Gimp, Paint, Thunderbird, Firefox and Scribes. It is easy to install and run alongside your existing operating system, so give it a try, because it is also easy to remove if you don't like it – which is unlikely.

Advantages of Linux

1. It is free
2. It is portable to any hardware platform.
3. It is secure and versatile
4. It is scalable
5. It is VIRUS free OS

Why use Linux?

This is one of the questions that most people ask. Why bother learning a completely different computing environment, when the operating system that ships with most desktops, laptops, and servers works just fine?

To answer that question, I would pose another question. Does that operating system you're currently using really work "just fine"? Or are you constantly battling viruses, malware, slowdowns, crashes, costly repairs, and licensing fees?

If you struggle with the above, and want to free yourself from the constant fear of losing data or having to take your computer in for the "yearly clean up," Linux might be the perfect platform for you. Linux has evolved into one of the most

reliable computer ecosystems on the planet. Combine that reliability with zero cost of entry and you have the perfect solution for a desktop platform.

That's right, zero cost of entry...as in free. You can install Linux on as many computers as you like without paying any money for software or server licensing (including costly Microsoft Client Access License – CALs). Working with Linux is a dream comes true. No more daily babysitting servers. In fact, Linux is as close to "set it and forget it" as you will ever find. And, on the off chance, one service on the server requires restarting; re-configuring, upgrading, etc… most likely the rest of the server won't be affected.

The Linux desktop or a server, if zero cost isn't enough to win you over – what about having an operating system that will work, trouble free, for as long as you use it? It is not once have an issue with malware, viruses, or random computer slow-down and server reboots? Only if the kernel is want to update. It is not out of the ordinary for a Linux server to go years without being rebooted. That's stability and dependability.

Linux is also distributed under an open source license. Open source follows the following key philosophies:

- The freedom to run the program, for any purpose.
- The freedom to study how the program works, and change it to make it do what you wish.
- The freedom to redistribute copies so you can help your neighbor.
- The freedom to distribute copies of your modified versions to others.

3.2 History of Linux

1969

All modern operating systems have their roots in 1969 when Dennis Ritchie and Ken Thompson developed the C language and the UNIX operating system at AT&T Bell Labs. They shared their source code (yes, there was open source back in the Seventies) with the rest of the world, including the hippies in Berkeley California. By 1975, when AT&T started selling UNIX commercially, about half of the source code was written by others. The hippies were not happy that a commercial company sold software that they had written; the resulting (legal) battle ended in there being two versions of UNIX: the official AT&T UNIX, and the free BSD Unix.

Development of BSD descendants like FreeBSD, Opens, Nets, Dragonfly BSD and PC-BSD is still active today.

https://en.wikipedia.org/wiki/Dennis_Ritchie
https://en.wikipedia.org/wiki/Ken_Thompson
https://en.wikipedia.org/wiki/BSD
https://en.wikipedia.org/wiki/Comparison_of_BSD_operating_systems

1980

In the Eighties many companies started developing their own UNIX: IBM created AIX, Sun SunOS (later Solaris), HP HP-UX and about a dozen other companies did the same. The result was a mess of UNIX dialects and a dozen different ways to do the same thing. And here is the first real root of Linux, when Richard Stallman aimed to end this era of UNIX separation and everybody re-inventing the wheel by starting the GNU project (GNU is Not Unix). His goal was to make an operating system that was freely available to everyone, and where everyone could work together (like in the Seventies). Many of the command line tools that you use today on Linux are GNU tools.

https://en.wikipedia.org/wiki/Richard_Stallman
https://en.wikipedia.org/wiki/IBM_AIX
https://en.wikipedia.org/wiki/HP-UX

1990

The Nineties started with Linus Torvalds, a Swedish speaking Finnish student, buying a386 computer and writing a brand new POSIX compliant kernel. He put the source code online, thinking it would never support anything but 386 hardware. Many people embraced the combination of this kernel with the GNU tools, and the rest, as they say, is history.

http://en.wikipedia.org/wiki/Linus_Torvalds
https://en.wikipedia.org/wiki/History_of_Linux
https://en.wikipedia.org/wiki/Linux
https://lwn.net

2015

Today more than 97 percent of the world's supercomputers (including the complete top 10), more than 80 percent of all smartphones, many millions of desktop computers, around 70 percent of all web servers, a large chunk of tablet computers, and several appliances (dvd-players, washing machines, dsl modems, routers, self-driving cars, space station laptops...) run Linux. Linux is by far the most commonly used operating system in the world.

Linux kernel version 4.0 was released in April 2015. Its source code grew by several hundred thousand lines (compared to version 3.19 from February 2015) thanks to contributions of thousands of developers paid by hundreds of

commercial companies including Red Hat, Intel, Samsung, Broadcom, Texas Instruments, IBM, Novell, Qualcomm, Nokia, Oracle, Google, AMD and even Microsoft (and many more).

http://kernelnewbies.org/DevelopmentStatistics
http://kernel.org
http://www.top500.org

In order to understand the popularity of Linux, we need to travel back in time, about 30 years ago. Imagine computers as big as houses, even stadiums. While the sizes of those computers posed substantial problems, there was one thing that made this even worse: every computer had a different operating system. Software was always customized to serve a specific purpose, and software for one given system didn't run on another system. Being able to work with one system didn't automatically mean that you could work with another. It was difficult, both for the users and the system administrators.

Computers were extremely expensive then, and sacrifices had to be made even after the original purchase just to get the users to understand how they worked. The total cost per unit of computing power was enormous.

Technologically the world was not quite that advanced, so they had to live with the size for another decade. In 1969, a team of developers in the Bell Labs laboratories started working on a solution for the software problem, to address these compatibility issues. They developed a new operating system, which was

- Simple and elegant.
- Written in the C programming language.
- Able to recycle code.

The Bell Labs developers named their project "UNIX." The code recycling features were very important. Until then, all commercially available computer systems were written in a code specifically developed for one system. UNIX on the other hand needed only a small piece of that special code, which is now commonly named the kernel. This kernel is the only piece of code that needs to be adapted for every specific system and forms the base of the UNIX system. The operating system and all other functions were built around this kernel and written in a higher programming language, C. This language was especially developed for creating the UNIX system. Using this new technique, it was much easier to develop an operating system that could run on many different types of hardware.

The software vendors were quick to adapt, since they could sell ten times more software almost effortlessly. Weird new situations came in existence: imagine for instance computers from different vendors communicating in the same network, or users working on different systems without the need for extra education to use another computer. UNIX did a great deal to help users become compatible with different systems.

Throughout the next couple of decades the development of UNIX continued. More things became possible to do and more hardware and the software vendors added support for UNIX to their products.

UNIX was initially found only in very large environments with mainframes and minicomputers (note that a PC is a "micro" computer). You had to work at a university, for the government or for large financial corporations in order to get your hands on a UNIX system.

But smaller computers were being developed, and by the end of the 80's, many people had home computers. By that time, there were several versions of UNIX available for the PC architecture, but none of them were truly free and more important: they were all terribly slow, so most people ran MS DOS or Windows 3.1 on their home PCs.

Linus and Linux

By the beginning of the 90s home PCs were finally powerful enough to run a full-blown UNIX. Linus Torvalds, a young man studying computer science at the University of Helsinki, thought it would be a good idea to have some sort of freely available academic version of UNIX, and promptly started to code.

He started to ask questions, looking for answers and solutions that would help him get UNIX on his PC. Below is one of his first posts in comp.os.minix, dating from 1991:

> From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
> Newsgroups: comp.os.minix
> Subject: Gcc-1.40 and a posix-question
> Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>
> Date: 3 Jul 91 10:00:50 GMT
> Hello netlanders,
>
> Due to a project I'm working on (in minix), I'm interested in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice.

From the start, it was Linus' goal to have a free system that was completely compliant with the original UNIX. That is why he asked for POSIX standards, POSIX still being the standard for UNIX.

In those days plug-and-play wasn't invented yet, but so many people were interested in having a UNIX system of their own, that this was only a small obstacle. New drivers became available for all kinds of new hardware, at a continuously rising speed. Almost as soon as a new piece of hardware became available, someone bought it and submitted it to the Linux test, as the system was gradually being called, releasing more free code for an ever wider range of hardware. These coders didn't stop at their PC's; every piece of hardware they could find was useful for Linux.

Back then, those people were called "nerds" or "freaks", but it didn't matter to them, as long as the supported hardware list grew longer and longer. Thanks to these people, Linux is now not only ideal to run on new PC's, but is also the system of choice for old and exotic hardware that would be useless if Linux didn't exist.

Two years after Linus' post, there were 12000 Linux users. The project, popular with hobbyists, grew steadily, all the while staying within the bounds of the POSIX standard. All the features of UNIX were added over the next couple of years, resulting in the mature operating system Linux has become today. Linux is a full UNIX clone, fit for use on workstations as well as on middle-range and high-end servers. Today, a lot of the important players on the hardware and software market each have their team of Linux developers; at your local dealer's you can even buy pre-installed Linux systems with official support - even though there is still a lot of hardware and software that is not supported, too.

3.3 Distribution of Linux

Linux has a number of different versions to suit nearly any type of user. From new users to hard-core users, you will find a “flavor” of Linux to match your needs. These versions are called distributions (or, in the short form, “distros.”) Nearly every distribution of Linux can be downloaded for free, burned onto disk (or USB thumb drive), and installed on as many machines as you like.

The most popular **Linux distributions are**:

1. Ubuntu Linux
2. Fedora
3. Debian
4. OpenSUSE
5. Linux Mint
6. Arch Linux
7. Deepin

Each distribution has a different take on the desktop. Some are for very modern user interfaces (such as Ubuntu's Unity, above, and Deepin's Deepin Desktop), whereas others stick with a more traditional desktop environment (openSUSE uses K Desktop Environment (KDE).

Find the Best Linux Desktop for You

With Linux comes choice. Along with that choice, comes debate. Which desktop is the best? Which offers the user-friendliest experience? The questions are not only never-ending, but date back over a decade where the "war" between KDE, GNOME, and every other desktop was given voice. It would, contend, however, that there is a desktop for every kind of user to be found within the Linux landscape. To that end, It want to take some of the most popular desktops and match them to end users.

There are no hard and fast rules, tests to take, or wizards to walk you through to your final Linux desktop destination. For most people it's about taste and features. But if you look at each desktop long enough, you discover there is a clear connection between desktop and end user. You can examine the following **Linux desktops**:

- Ubuntu Unity
- GNOME 3
- Cinnamon
- KDE
- Enlightenment
- XFCE
- Deepin Desktop

Each of the above desktops has a strong following (with good reason). As well, each desktop offers a wholly unique experience with plenty of features to please anyone. However -- getting connected with the right desktop, up front,

can go a very long way to ensuring an overall positive experience with the Linux desktop. With that said, let's connect users with desktops.

You can check out the top 100 distributions on the http://www.distrowatch.com site. And don't think the **server** has been left behind. For this arena, you can turn to:

- Red Hat Enterprise Linux
- Ubuntu Server
- CentOS
- SUSE Enterprise Linux.

Some of the above server distributions are free (such as Ubuntu Server and CentOS) and some have an associated price (such as Red Hat Enterprise Linux and SUSE Enterprise Linux). Those with an associated price also include support.

Ubuntu

Ubuntu is probably the most well-known Linux distribution. Ubuntu is based on Debian, but it has its own software repositories. Much of the software in these repositories is synced from Debian's repositories.

The Ubuntu project has a focus on providing a solid desktop (and server) experience, and it isn't afraid to build its own custom technology to do it. Ubuntu used to use the GNOME 2 desktop environment, but it now uses its own Unity desktop environment. Ubuntu is even building its own Mir graphical server while other distributions are working on the Wayland.

Ubuntu is modern without being too bleeding edge. It offers releases every six months, with a more stable LTS (Long Term Support) release every

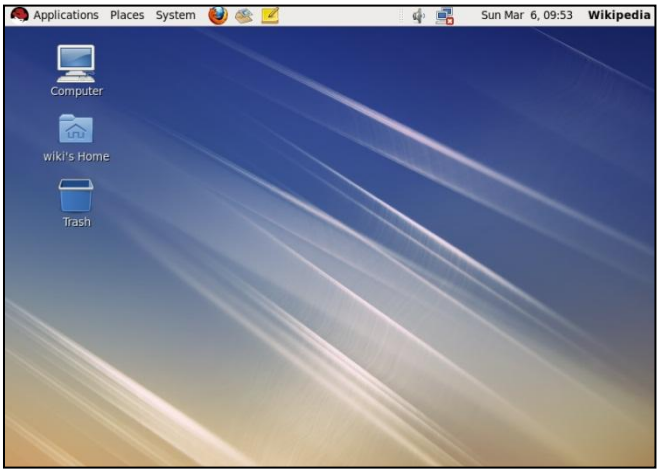
two years. Ubuntu is currently working on expanding the Ubuntu distribution to run on smart phones and tablets.

Fedora

Fedora is a project with a strong focus on free software — you won't find an easy way to install proprietary graphics drivers here, although third-party repositories are available. Fedora is bleeding edge and contains the latest versions of software. Unlike Ubuntu, Fedora doesn't make its own desktop environment or other software. Instead, the Fedora project uses "upstream" software, providing a platform that integrates all this upstream software without adding their own custom tools or patching it too much. Fedora comes with the GNOME 3 desktop environment by default, although you can also get "spins" that come with other desktop environments.

Fedora is sponsored by Red Hat, and is the foundation for the commercial Red Hat Enterprise Linux project. Unlike RHEL, Fedora is bleeding edge and not supported for long. If you want a more stable release that's supported for longer, Red Hat would prefer you use their Enterprise product.

Red Hat Enterprise Linux



Red Hat Enterprise Linux is a commercial Linux distribution intended for servers and workstations. It's based on the open-source Fedora project, but is designed to be a stable platform with long-term support.

Red Hat uses trademark law to prevent their official Red Hat Enterprise Linux software from being redistributed. However, the core software is free and open-source. Cento is a community project that takes the Red Hat Enterprise Linux code, removes all Red Hat's trademarks, and makes it available for free use and distribution. It's a free version. If you want a stable platform that will be supported for a long time, Red Hat recently announced they're collaborating, so CentOS is now part of Red Hat itself.

Open Suse / Suse Linux Enterprise

Open Suse is a community-created Linux distribution sponsored by Novell. Novell purchased Suse Linux in 2003, and they still create an enterprise Linux project known as Suse Linux Enterprise. Where Red Hat has the Fedora project that feeds into Red Hat Enterprise Linux, Novell has the Opens use project that feeds into Suse Linux Enterprise. Like Fedora, OpenSuse is a more bleeding edge version of Linux. Suse was once one of the great user-friendly desktop Linux distributions, but Ubuntu eventually took that crown.

Debian

Debian is an operating system composed only of free, open-source software. The Debian project has been operating since 1993 — over 20 years ago! This widely respected project is still releasing new versions of Debian, but it's known for moving much more slowly than distributions like Ubuntu or Linux Mint. This can make it more stable and conservative, which is ideal for some systems.

Ubuntu was originally founded to take the core bits of stable Debian and improve on them more quickly, packaging the software together into a user-friendly system that's more frequently updated.

Linux Mint

Mint is a Linux distribution built on top of Ubuntu. It uses Ubuntu's software repositories, so the same packages are available on both. Originally, Mint was an alternative distribution loved mainly because it included media codecs and proprietary software that Ubuntu didn't include by default.

This distribution now has its own identity. You won't find Ubuntu's own Unity desktop here — instead, you get a more traditional Cinnamon or MATE desktop. Mint takes a more relaxed approach to software updates and won't automatically install critical software updates. Controversially, this has led some Ubuntu developers to label it insecure.

Others

Distributions like CentOS, Oracle Enterprise Linux and Scientific Linux are based on Red Hat Enterprise Linux and share many of the same principles, directories and system administration techniques. Linux Mint, Edubuntu and many other *buntu named distributions are based on Ubuntu and thus share a lot with Debian. There are hundreds of other Linux distributions.

Check your progress 1

Q.1 What is Linux ? What are the various flavors of Linux ? What are the advantages of Linux ?
A._____

_____

_____

3.4 Devices and drivers used

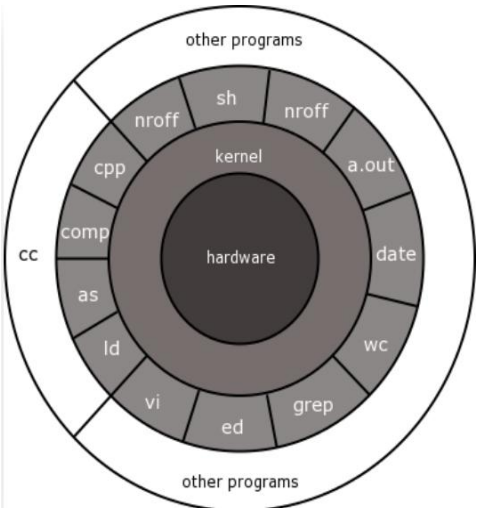
The CPU is not the only intelligent device in the systems. Every physical device has its own hardware controller. The keyboard, mouse and serial ports are controlled by a Super IO (Input Output) chip, the Integrated Drive Electronics (IDE) disks by an IDE controller, Small Computer System Interface (SCSI) disks by a SCSI controller and so on. Each hardware controller has its own control and status registers Certificate Signing Request (CSRs) and these differ between devices. The CSRs for an Adaptec 2940 SCSI controller are completely different from those of an NCR 810 SCSI controller. The CSRs are used to start and stop the device, to initialize it and to diagnose any problems with it. Instead of putting code to manage the hardware controllers in the system into every application, the code is kept in the Linux kernel. The kernel is the essential center of a computer operating system, the core that provides basic services for all other parts of the operating system. A synonym is nucleus. A kernel can be contrasted with a shell, the outermost part of an operating system that interacts with user commands. The software that handles or manages a hardware controller is known as a device driver. The Linux kernel device drivers are, essentially, a shared library of privileged, memory resident, low-level hardware handling routines. It is Linux's device drivers that handle the peculiarities of the devices they are managing.
There are many different device drivers in the Linux kernel (that is one of Linux's strengths) but they all share some common attributes:

1. Kernel Code

Device drivers are part of the kernel and, like other code within the kernel, if they go wrong they can seriously damage the system. A badly written driver may even crash the system, possibly corrupting file systems and losing data,

2. Kernel interfaces

Device drivers must provide a standard interface to the Linux kernel or to the subsystem that they are part of. For example, the terminal driver provides a file IO interface to the Linux kernel and a SCSI device driver. Which provides a SCSI device interface to the SCSI subsystem in turn provides both file IO and buffer cache interfaces to the kernel.



3. Kernel mechanisms and services

Device drivers make use of standard kernel services such as memory allocation, interrupt delivery and wait queues to operate.

4. Loadable

Most of the Linux device drivers can be loaded on demand as kernel modules when they are needed and unloaded when they are no longer being used. This makes the kernel very adaptable and efficient with the system's resources.

5. Configurable

Linux device drivers can be built into the kernel. Which devices are built is configurable when the kernel is compiled**,**

6. Dynamic

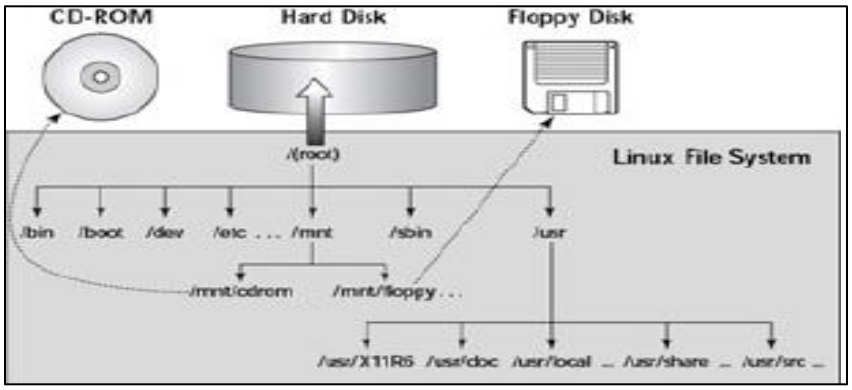
As the system boots and each device driver is initialized it looks for the hardware devices that it is controlling. It does not matter if the device being controlled by a particular device driver does not exist. In this case the device driver is simply redundant and causes no harm apart from occupying a little of the system's memory.

3.5. File system hierarchy

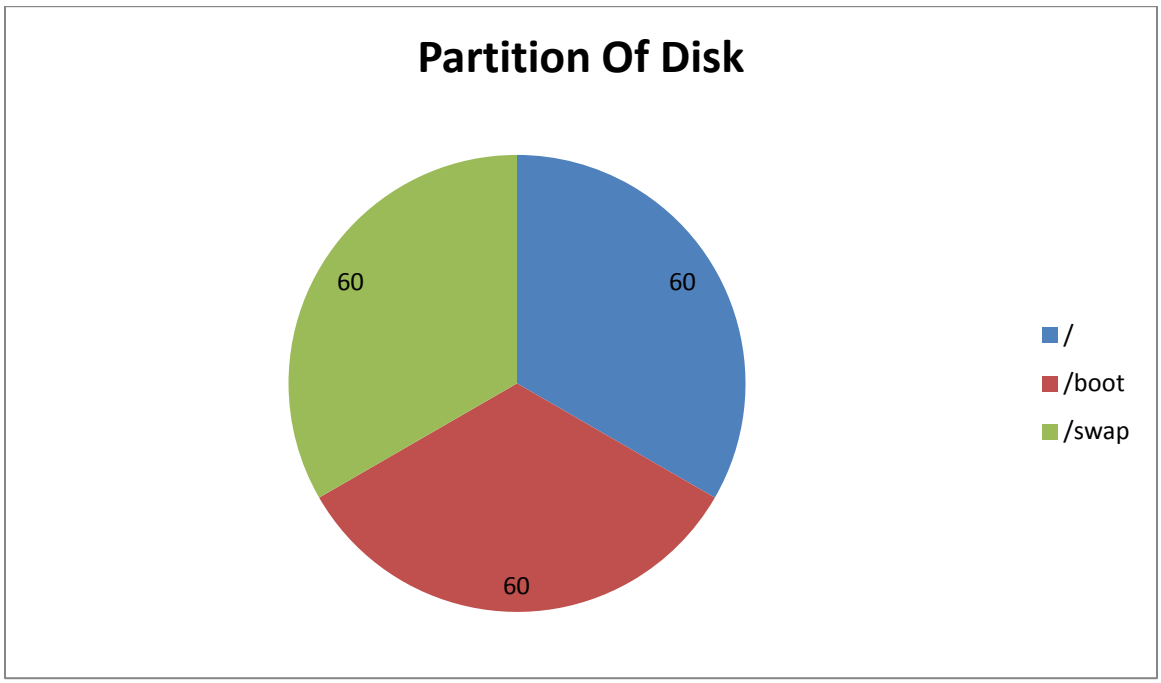
Like any other operating system, Linux organizes information in files and directories. A directory is a special file that can contain other files and directories. Because a directory can contain other directories, this method of organizing files gives rise to a hierarchical structure. This hierarchical organization of files is the file system.

Linux follows the file system hierarchy system. Each directory that is inbuilt in Linux is prepared for a particular reason. The files in these directories carry out the functions of the Linux. The top level directory is root directory. The root is identified by the symbol /.The /boot directory comprises of the OS kernel along with files used during boot strap process. The /swap directory supports virtual memory. Data is written in the swap partition when there is less memory in the RAM. Linux during installation has to be installed in the un partitioned disk. Linux takes care of partitioning the disk into 3 partitions. these are the / or root,/boot,/swap. The /boot takes 200mb and ./ or root takes 5gb of hard disk. The /swap takes twice of memory, if i have 2 GB RAM 4GB of HDD is used. The /root directory is the root home directory. It provides the working environment of the root user. /home is the directory for other users./etc folder contains start up and shut down shell script used to start and stop individual programs./usr is the folder that holds default softwares./opt is the optional directory for the /usr which holds 3rd partysoft wares./SBIN comprises of commands used by super user./bin contains commands used by normal user./dev contains device files.

/proc comprises of process files./var comprises of variable files./lib comprises of library files./opt comprises of optional files./mnt is the mount directory./media comprises of removable devises such as pen drive./srv comprises of service data./proc comprises of processing information./foundlost comprises of accidentally deleted files for recovery.



During the time of installation Linux is to be installed on an un partitioned disk. Linux partitions un partitioned disk into 3 partitions



If you're familiar with other operating systems such as Windows, you may find something missing in the Linux file system: You don't find drive letters in Linux. All disk drives and CD-ROM drives are part of a single file system.

In Linux, you can have long filenames (up to 256 characters), and filenames are case sensitive. Often these filenames have multiple extensions, such as sample.tar.Z. Unix filenames can take many forms, such as the following: index.html, Makefile, binutils 2.15.92.0.2-5.i386.rpm, vsftpd_ 2.0.3-1_i386.deb, .bash_profile, and httpd_src.tar.gz.

To locate a file, you need more than just the filename. You also need information about the directory hierarchy. The extended filename, showing the full hierarchy of directories leading to the file, is the pathname. As the name implies, it's the path to the file through the maze of the file system.

Shows a typical pathname for a file in Linux.

Directories in the Linux File System

Directory	Content
/	Base of the file system. All files and directories are contained logically in the root, or /, directory, regardless of their physical locations.
/Bin	Executable programs that are part of the Linux operating system. Many Linux commands, such as cat, cp, ls, more, and tar, are located in /bin.
/boot	Linux kernel and other files that the LILO and GRUB boot managers need. (The kernel and other files can be anywhere, but placing them in the /boot directory is customary.
/dev	Special files that represent devices attached to the system.
/etc	Most system configuration files and the initialization scripts (in the /etc/rc.d subdirectory).
/home	Home directories of all users. User ramus's home directory, for example, is /home/ramu
/Lib	Library files for all programs stored in /sbin and /bin directories (including the loadable driver modules) needed to start Linux.

/lost+found	Lost files. Every disk partition has a lost+found directory.
/media	The /media/cd rom or /media/cdrom0 directory is for mounting the CD/DVD-ROM drive. If you have a CD/DVD recorder, you find a /media/cd recorder directory instead of /media/cd rom and may also find /media/ DVD. Used for mounting file systems on removable media, such as CD/DVD-ROM drives, flash drives, external drives, floppy disks, and Zip drives. If you have a very old machine that still has a floppy drive on it, then the /media/floppy directory will also exist for mounting floppy disks. Temporarily mounted file systems.
/opt	Storage for large application software packages. For example, some distributions install the LibreOffice.org Office suite in the/opt directory. Various information about the processes running in the Linux System.
/root	Home directory for the root user.
/sbin	Executable files representing commands typically used for system administration tasks and used by the root user. Commands such as halt and shutdown reside in the /sbin directory.
/srv	Data for services (such as web and FTP) offered by this system.
/sys	Information about the devices, as seen by the Linux kernel.
/temp	Temporary directory that any user can use as a scratch directory, meaning that the contents of this directory are considered unimportant and usually are deleted every time the system boots.
/usr	Subdirectories for many important programs,

	such as the X Window System (in the /usr/X11R6 directory) and the online manual.
/var	Various system files (such as logs), as well as directories for holding other information, such as files for the web server and anonymous FTP server.

3.6. The Components: Kernel, Distribution, XFree86, Sawfish, Gnome

Kernel

Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It is consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.

Distribution

Linux comes in broadly 2 kinds of distributions. These are desk top distribution and the server side distribution. The desktop distribution enable us to load Linux at the desktop and the server version enables us to load Linux at the Server. Some of the desktop distributions are Fedora, Ubuntu and that at the server are the Red Hat Enterprise versions.

XFree86

XFree86 is free version of Unix X windows system.it provides graphical interface for users. It uses a client server model which means that the main operating system works at one machine and the rest of the systems communicate to it through sessions. By this client systems that are available locally can communicate with the server and those that are very far away or remote systems communicate with the server. This enables data security and high speed transmission of data. Data is located at one place which is safe and secure. The x free 86 graphical interface enables ease of access such a file creation, modification and other features for the authorized user of the system. It comes in two versions 3x and 4x.The configuration files of these two versions are easy to use. The configuration means server drivers, mouse ,keyboard, fonts and display configuration such as configuration modes and resolutions. There are number of tools that come with XFree86.The list of tools are

- XF86 config which is a text based configuration file

- XF86 setup graphical which helps in configuration
- XF86 cfg which allows to graphically setup the configuration

The Red Hat has a XConfigurator that allows to change graphical environment
The Suse Linux uses SaX2 to configure the Linux environment.

For detail study follow the link :-https://youtu.be/kAiwUVH7BWM

Sawfish

Sawfish is a window manager for the X Window System. It aims to manage windows in the most flexible and attractive manner possible. It matches a created window by multiple criteria, and automatically alters the window's position, sets the software theme per-window, or removes the borders entirely. There is a GUI configuration utility for users who do not wish to edit configuration files directly.

Sawfish is an extensible window manager using an Emacs Lisp-like scripting language. All window decorations are configurable and all user-interface policy is controlled through the Lisp language. But Sawfish also comes with a GUI configurator which allows basic customization (including switching themes) without having to know any Lisp.

Sawfish focuses mainly on its core ability to manage windows, and not on all the extras like application docks and modules that many modern window managers provide. This means that Sawfish works really well when used in conjunction with a desktop environment, particularly GNOME, as that provides just the facilities that Sawfish lacks. In fact, Sawfish was once the default window manager for the GNOME desktop.

Gnome

Gnome is part of the GNU project and part of the free software, or open source, movement. **Gnome** is a Windows-like desktop system that works on UNIX and UNIX-like systems and is not dependent on any one window manager. The current version runs on **Linux**, FreeBSD, IRIX and Solaris.

The Gnome (pronounced "Gah-NOME") project's aim is to build a complete, user-friendly desktop based entirely on free software. It is not a window manager, and in fact has to be run in conjunction with a window manager. GNOME has advanced rapidly and is now very popular, particularly on the Linux platform. GNOME has to be run alongside a traditional window manager. This is typically Sawfish, which is in fact shipped with GNOME. Any window manager can be used, but if that window manager is not GNOME-aware, you will lose some functionality. Enlightenment and Icewm are other GNOME-aware window managers. Gnome can be installed by using **YUM** which is used to install software packages. We shall be discussing about **YUM** shortly. For installation procedures please follow the link **https://www.youtube.com/watch?v=3EkldXVm34w**

3.7. The command-line commands

The command line commands are the commands that are issued at the terminal Mode. The command line commands help the user to run commands to list files, make Folders, copy files, delete files, hide files etc. It also helps the user to load, update, delete Software Packages by giving commands. People those who mastered commands prefer command line commands. Example of command line command is # cd ~ .this command takes the user to the root home directory

What is the shell?

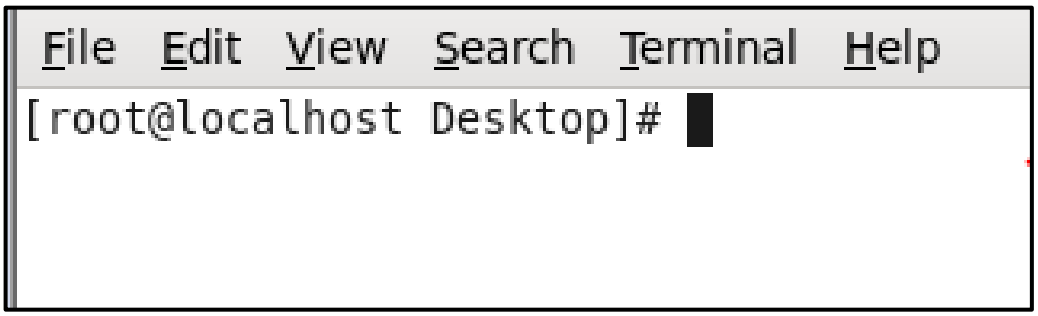
Shell is a user program or it's environment provided for user interaction. Shell is an command language interpreter that executes commands read from the standard input device (keyboard) or from a file.Shell is not part of system kernel, but uses the system kernel to execute programs, create files etc. Shell in Linux is Bash shell.

Terminal Mode

Click on open in terminal mode to open Linux in terminal mode as shown below.

:- Stands for super user

$:- Stands for normal user.



Basic Linux commands used in terminal Mode

Basic Linux commands.

$ls	It will display the list contents in a folder
$clear	It will clear the screen
$mkdir	It creates a new directory
$cat >	(new file name) It will create a new file , for saving the file you have to press ctrl + d
$cat	(existing file name)It will display the contents of a file
$rm	(file name) It will delete a file
$rm – rf	(dir name). It will delete a directory
$mv	(old file name) (new file name).It will rename a file (old directory name) (new directory name).It will rename a directory
$mv	(source file name)(target directory name).It will move the file from one location to the other.
$touch	(new file name) It will create an empty file
$date	It will display the current system date
$man	(command) it will display the mnuals and syntax of the command.
$halt	To shut down the Linux environment

Our first keystrokes

So let's get started. Launch the terminal emulator! Once it comes up, you should see something like this:

[Sandy@Linuxbox ~]$

This is called a shell prompt, and it appears whenever the shell is ready to accept input. While it may vary in appearance somewhat, depending on the distribution, it will usually include your

username@machinename, followed by the current working directory (more about that in a little bit) and a dol-lar sign.

If the last character of the prompt is a hash mark (#) rather than a dollar sign, the terminal session has superuser privileges. This means that either we are logged in as the root user or we've selected a terminal emulator that provides superuser (administrative) privileges.

[Sandy@Linuxbox ~]$ xyz

Since this command makes no sense, the shell tells us so and gives us another chance:bash: **xyz**: command not found [Sandy@Linuxbox ~] $

Some Simple Commands

DATE

This command displays the current time and date.
[Sandy@Linuxbox ~]$ date↵
Fri Jul 01 00:20:45 IST 2016

CAL

This command used to displays a calendar of the current month.
[Sandy@Linuxbox ~]$ cal ↵

July 2016

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

<u>List of A-Z Index of the Bash command line for Linux.</u>

```
A
alias    Create an alias •
  apropos  Search Help manual pages (man -k)
apt-get  Search for and install software packages
(Debian/Ubuntu)
aptitude Search for and install software packages
(Debian/Ubuntu)
aspell   Spell Checker
awk      Find and Replace text, database
sort/validate/index
B
```

basename Strip directory and suffix from filenames
bash GNU Bourne-Again Shell
bc Arbitrary precision calculator language
bg Send to background
break Exit from a loop •
builtin Run a shell built in
bzip2 Compress or decompress named file(s)
C
cal Display a calendar
case Conditionally perform a command
cat Concatenate and print (display) the content of files
cd Change Directory
cfdisk Partition table manipulator for Linux
chgrp Change group ownership
chmod Change access permissions
chown Change file owner and group
chroot Run a command with a different root directory
chkconfig System services (runlevel)
cksum Print CRC checksum and byte counts
clear Clears terminal screen
cmp Compare two files
comm Compare two sorted files line by line
command Run a command - ignoring shell functions •
continue Resume the next iteration of a loop •
cp Copy one or more files to another location
cron Daemon to execute scheduled commands
crontab Schedule a command to run at a later time
csplit Split a file into context-determined pieces
curl Transfer data from or to a server
cut Divide a file into several parts

D
date Display or change the date & time
dc Desk Calculator
dd Convert and copy a file, write disk headers, boot records
ddrescue Data recovery tool
declare Declare variables and give them attributes •
df Display free disk space
diff Display the differences between two files
diff3 Show differences among three files
dig DNS lookup
dir Briefly list directory contents
dircolors Colour setup for `ls'

dirname Convert a full pathname to just a path
dirs Display list of remembered directories
dmesg Print kernel & driver messages
du Estimate file space usage
E
echo Display message on screen •
egrep Search file(s) for lines that match an extended expression
eject Eject removable media
enable Enable and disable built in shell commands •
env Environment variables
ethtool Ethernet card settings
eval Evaluate several commands/arguments
exec Execute a command
exit Exit the shell
expectAutomate arbitrary applications accessed over a terminal
expand Convert tabs to spaces
export Set an environment variable
expr Evaluate expressions
F
false Do nothing, unsuccessfully
fdformat Low-level format a floppy disk
fdisk Partition table manipulator for Linux
fg Send job to foreground
fgrep Search file(s) for lines that match a fixed string
file Determine file type
find Search for files that meet a desired criteria
fmt Reformat paragraph text
fold Wrap text to fit a specified width.
for Expand *words*, and execute *commands*
format Format disks or tapes
free Display memory usage
fsck File system consistency check and repair
ftp File Transfer Protocol
function Define Function Macros
fuser Identify/kill the process that is accessing a file
G
gawk Find and Replace text within file(s)
getopts Parse positional parameters
grep Search file(s) for lines that match a given pattern
groupadd Add a user security group
groupdel Delete a group

groupmod Modify a group
groups Print group names a user is in
gzip Compress or decompress named file(s)
H
hash Remember the full pathname of a name argument
head Output the first part of file(s)
help Display help for a built-in command •
history Command History
hostname Print or set system name
htop Interactive process viewer
I
iconv Convert the character set of a file
id Print user and group id's
if Conditionally perform a command
ifconfig Configure a network interface
ifdown Stop a network interface
ifup Start a network interface up
import Capture an X server screen and save the image
to file
install Copy files and set attributes
ip Routing, devices and tunnels

J
jobs List active jobs •
join Join lines on a common field
K
kill Kill a process by specifying its PID
killall Kill processes by name

L
less Display output one screen at a time
let Perform arithmetic on shell variables •
link Create a link to a file
ln Create a symbolic link to a file
local Create variables •
locate Find files
logname Print current login name
logout Exit a login shell •
look Display lines beginning with a given string
lpc Line printer control program
lpr Off line print
lprint Print a file
lprintd Abort a print job
lprintq List the print queue
lprm Remove jobs from the print queue
ls List information about file(s)
lsof List open files

M
make Recompile a group of programs
man Help manual
mkdir Create new folder(s)
mkfifo Make FIFOs (named pipes)
mkisofs Create an hybrid ISO9660/JOLIET/HFS filesystem
mknod Make block or character special files
more Display output one screen at a time
most Browse or page through a text file
mount Mount a file system
mtools Manipulate MS-DOS files
mtr Network diagnostics (traceroute/ping)
mv Move or rename files or directories
mmv Mass Move and rename (files)
N
nc Netcat, read and write data across networks
netstat Networking information
nice Set the priority of a command or job
nl Number lines and write files
nohup Run a command immune to hangups
notify-send Send desktop notifications
nslookup Query Internet name servers interactively
O
open Open a file in its default application
op Operator access
P
passwd Modify a user password
paste Merge lines of files
pathchk Check file name portability
ping Test a network connection
pkill Kill processes by a full or partial name.
popd Restore the previous value of the current directory
pr Prepare files for printing
printcap Printer capability database
printenv Print environment variables
printf Format and print data •
ps Process status
pushd Save and then change the current directory
pv Monitor the progress of data through a pipe
pwd Print Working Directory
Q
quota Display disk usage and limits
quotacheck Scan a file system for disk usage
R
ram ram disk device

rar Archive files with compression
rcp Copy files between two machines
read Read a line from standard input •
readarray Read from stdin into an array variable •
readonly Mark variables/functions as readonly
reboot Reboot the system
rename Rename files
renice Alter priority of running processes
remsync Synchronize remote files via email
return Exit a shell function
rev Reverse lines of a file
rm Remove files
rmdir Remove folder(s)
rsync Remote file copy (Synchronize file trees)
S
screen Multiplex terminal, run remote shells via ssh
scp Secure copy (remote file copy)
sdiff Merge two files interactively
sed Stream Editor
select Accept keyboard input
seq Print numeric sequences
set Manipulate shell variables and functions
sftp Secure File Transfer Program
shift Shift positional parameters
shopt Shell Options
shutdown Shutdown or restart Linux
sleep Delay for a specified time
slocate Find files
sort Sort text files
source Run commands from a file '.'
split Split a file into fixed-size pieces
ssh Secure Shell client (remote login program)
stat Display file or file system status
strace Trace system calls and signals
su Substitute user identity
sudo Execute a command as another user
sum Print a checksum for a file
suspend Suspend execution of this shell •
sync Synchronize data on disk with memory
T
tail Output the last part of file
tar Store, list or extract files in an archive
tee Redirect output to multiple files
test Evaluate a conditional expression
time Measure Program running time
timeout Run a command with a time limit
times User and system times

touch Change file timestamps
top List processes running on the system
tput Set terminal-dependent capabilities, color, position
traceroute Trace Route to Host
trap Run a command when a signal is set(bourne)
tr Translate, squeeze, and/or delete characters
true Do nothing, successfully
tsort Topological sort
tty Print filename of terminal on stdin
type Describe a command •
U
ulimit Limit user resources •
umask Users file creation mask
umount Unmount a device
unalias Remove an alias •
uname Print system information
unexpand Convert spaces to tabs
uniq Uniquify files
units Convert units from one scale to another
unrar Extract files from a rar archive
unset Remove variable or function names
unshar Unpack shell archive scripts
until Execute commands (until error)
uptime Show uptime
useradd Create new user account
userdel Delete a user account
usermod Modify user account
users List users currently logged in
uuencode Encode a binary file
uudecode Decode a file created by uuencode
V
v Verbosely list directory contents (`ls -l -b')
vdir Verbosely list directory contents (`ls -l -b')
vi Text Editor
vmstat Report virtual memory statistics
W
wait Wait for a process to complete •
watch Execute/display a program periodically
wc Print byte, word, and line counts
whereis Search the user's $path, man pages and source files for a program
which Search the user's $path for a program file
while Execute commands
who Print all usernames currently logged in

```
whoami    Print the current user id and name (`id -un')
  wget    Retrieve web pages or files via HTTP, HTTPS
          or FTP
write     Send a message to another user
X
xargs     Execute utility, passing constructed argument
list(s)
xdg-open Open a file or URL in the user's preferred
application.
xz        Compress or decompress .xz and .lzma files
yes       Print a string until interrupted
zip       Package and compress (archive) files.
.         Run a command script in the current shell
!!        Run the last command again
###       Comment / Remark
```

Check your progress 2

Q.1. What is Kernel ?

A._____

_____

_____

Q.2. What is Gnome?

A._____

_____

_____

Q.3. What is XFree86?

A._____

_____

_____

Q.4. What are command line commands?

A._____

_____

_____

3.8 File management commands

The commands that are used to manage the files,folders and administer the Linux system are termed as File management commands

1. The tilde (~) symbol stands for your home directory. If you are *user*, then the tilde (~) stands for /home/*user*
2. **pwd**: The **pwd** command will allow you to know in which directory you're located (**pwd** stands for "print working directory"). Example:**"pwd"** in the Desktop directory will show "~/Desktop". Note that the GNOME Terminal also displays this information in the title bar of its window. A useful gnemonic is "present working directory."
3. **ls**: The **ls** command will show you ('list') the files in your current directory. Used with certain options, you can see sizes of files, when files were made, and permissions of files. Example: **"ls ~"** will show you the files that are in your home directory.
4. **cd**: The **cd** command will allow you to change directories. When you open a terminal you will be in your home directory. To move around the file system you will use **cd**. Examples:
   1. To navigate into the root directory, use **"cd /"**
   2. To navigate to your home directory, use **"cd"** or **"cd ~"**
   3. To navigate up one directory level, use **"cd .."**
   4. To navigate to the previous directory (or back), use **"cd -"**
   5. To navigate through multiple levels of directory at once, specify the full directory path that you want to go to. For example, use, **"cd /var/www"** to go directly to the /www subdirectory of /var/. As another example, **"cd ~/Desktop"** will move you to the Desktop subdirectory inside your home directory.
5. **cp**: The **cp** command will make a copy of a file for you. Example: **"cp file foo"** will make an exact copy of "file" and name it "foo", but the file "file" will still be there. If you are copying a directory, you must use **"cp -r directory foo"** (copy recursively). (To understand what "recursively" means, think of it this way: to copy the directory and all its files and

subdirectories and all their files and subdirectories of the subdirectories and all their files, and on and on, "recursively")

6. **mv**: The **mv** command will move a file to a different location or will rename a file. Examples are as follows: **"mv file foo"** will rename the file "file" to "foo". **"mv foo ~/Desktop"** will move the file "foo" to your Desktop directory, but it will not rename it. You must specify a new file name to rename a file.
    1. To save on typing, you can substitute '~' in place of the home directory.
    2. Note that if you are using **mv** with **sudo** you can use the ~ shortcut, because the terminal expands the ~ to your home directory. However, when you open a root shell with **sudo -i** or **sudo -s**, ~ will refer to the root account's home directory, not your own.
7. **rm**: Use this command to remove or delete a file in your directory.
8. **rmdir**: The **rmdir** command will delete an *empty* directory. To delete a directory and all of its contents recursively, use **rm -r** instead.
9. **mkdir**: The **mkdir** command will allow you to create directories. Example: **"mkdir music"** will create a directory called "music".
10. **man**: The **man** command is used to show you the manual of other commands. Try **"man man"** to get the man page for **man** itself. See the "**Man** & Getting Help" section down the page for more information.
11. **sudo**: The **sudo** command is used to perform file operations on files that the **Root User** would only be allowed to change. An example would be trying to move one of your documents that another user accidentally moved to / back to your **documents** directory. Normally, to move the file, you would type **mv /mydoc.odt ~/Documents/mydoc.odt**, but you are not allowed to modify files outside of your home directory. To get around this, you would type **sudo mv /mydoc.odt ~/Documents/mydoc.odt**. This will successfully move the file back to its correct location, provided that you are not a **standard** user, who has less (administrative) ability than an **administrator**. Be aware, though, that by using the **sudo** command, you need to be **extra** careful. It is easier to **damage** your system by using the **sudo** command. For more information about the **sudo** command.

File Commands

ls - directory listing
ls -al - formatted listing with hidden files
cd *dir* - change directory to *dir*
cd - change to home
pwd - show current directory
mkdir *dir* - create a directory *dir*
rm *file* - delete *file*
rm -r *dir* - delete directory *dir*
rm -f *file* - force remove *file*
rm - rf *dir* - force remove directory *dir* *
cp *filel file2*- copy *filel to file2*
cp -r *dirl dir2* - copy *dirl to dir2;* create *dir2* if it doesn't exist
my filel file2 - rename or move *filel* to *file2*
if *file2* is an existing directory, moves *filel* into directory *file2*
In *-s file link* - create symbolic link *link* to *file* touch *file* - create or update *file*
cat >*file* - places standard input into *file*
more *file* - output the contents of *file*
head *file* - output the first 10 lines of *file*
tail *file* - output the last 10 lines of *file*
tail -f *file* - output the contents of *file* as it grows, starting with the last 10 lines

3.9 Working with Nano

Nano is a text editor suited to working in UNIX. It is not as powerful as PC window-based editors, as it does not rely on the mouse, but still has many useful features. Most **nano** commands are invoked by holding down the Ctrl key (that is, the control key), and pressing one of the other keys.

Nano is a small, free and friendly editor which aims to replace Pico, the default editor included in the non-free Pine package. Rather than just copying Pico's look and feel, **nano** also implements some missing (or disabled by default) features in Pico, such as "search and replace" and "go to line number".

Sometimes, a graphical text editor like gedit or kate cannot be used (because you're in a virtual console for example). Luckily, there are text editors

for the terminal. An easy one is nano, but I cannot understand how to work with it.
If I start nano by running nano, the bottom text is supposed to help me but all I see are things like ^G Get Help ^O WriteOut.

1. How can I open text files for editing?

   This is the default in nano. Open and file and you're set to start editing:
   nano filename

   Note: you won't be able to save unless you have write permissions for that file.

2. How can I save the file?

   F3 will let you save without exiting. Otherwise, Ctrl + X will prompt you if you've made changes. Press Y when it asks, and Enter to confirm the filename.

3. How can I quit the editor without saving the changes?

   Ctrl + X, then N when it asks if you want to save.

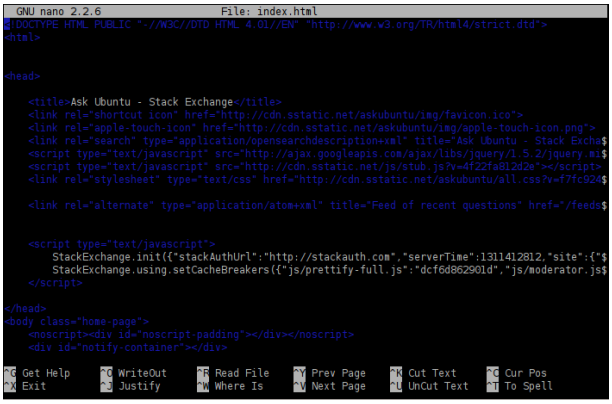
4. How to edit? I heard that you've to enter some commands to begin editing in vi, is this true for nano too?

   As above, no. nano is simple. It drops you in edit mode as soon as it opens. You can use arrow keys, Page Up / Page Down and Home / End as in gedit. You cannot use the mouse for moving the cursor position.

5. Sometimes, if I manage to open a file, the text is unreadable due to its colors. How can I disable these colors? (see the image below)

   Colours are loaded through the nanorc framework. These are files that are loaded when nano loads which basically spell out the syntax highlighting. To toggle syntax highlighting, press Alt + Y. To disable it permanently for certain file types, edit /etc/nanorc and put a hash mark (#) before include "/usr/share/nano/*.nanorc".

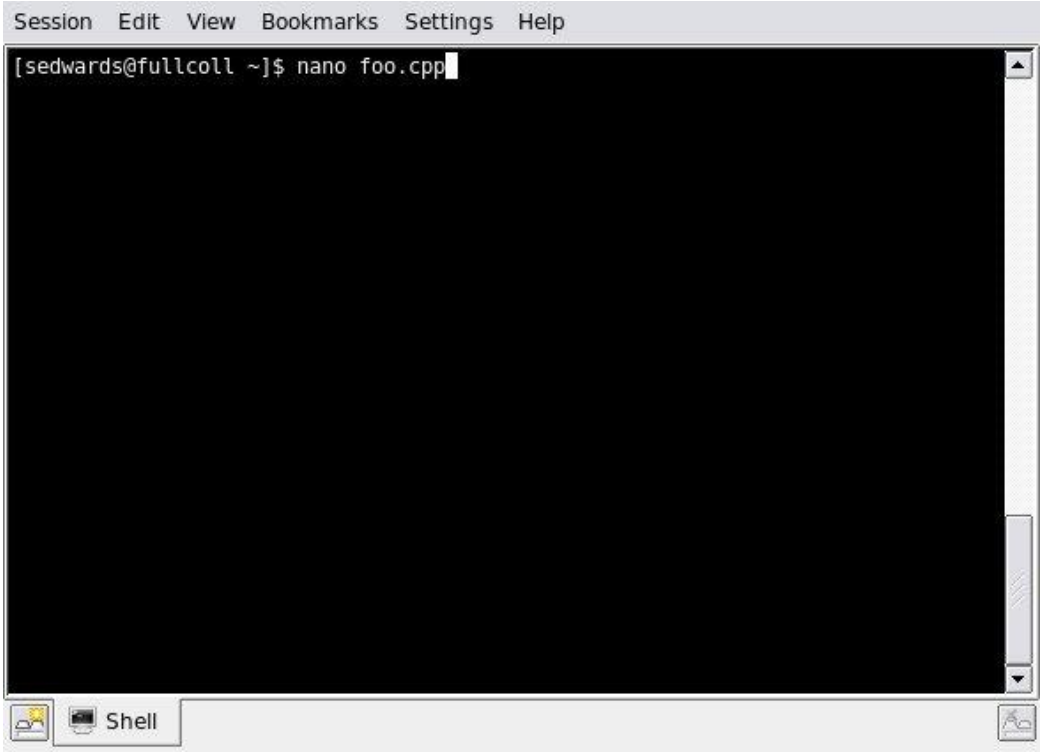
6. In the some files, lines are truncated because those do not fit in the screen. How can I prevent that from happening? (see the image below)



Well I've been trying to find something but the best I could see was enabling soft-line-wrap with the funky key-combination of: Alt + $ (Alt + Shift + 4). To enable soft line wrapping by default, add the below line to ~/.nanorc:

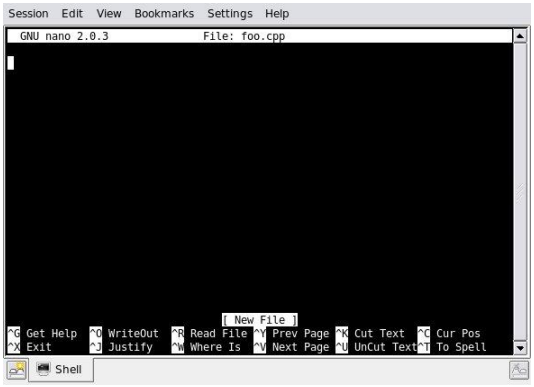
set softwrap

The example below shows that, if you wish, you may also provide the name of the file you want to edit:



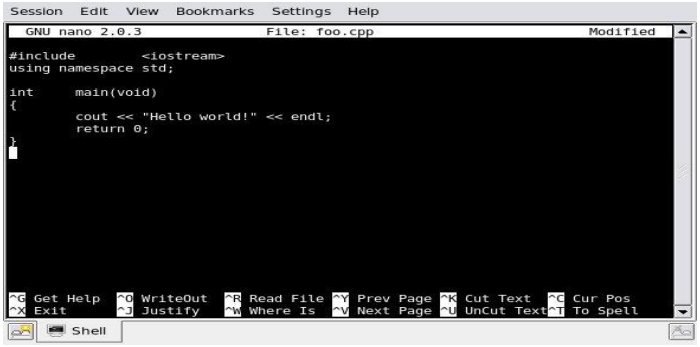
After pressing the **Enter** key, the nano editor appears (see image below). Notice the following elements:

- The top line displays the version of nano in the left corner and the name of the file being edited.
- The 3rd line from the bottom indicates the status of the file you're editing; in the image below it shows that foo.cpp is a “New File”.
- The last two lines of the screen present a menu of useful keyboard commands. For example, **^X** means to press **Ctrl+x** will exit the nano text editor. These aren't the only commands available, to see an entire list of commands enter **Ctrl+g**, will bring up the help window.

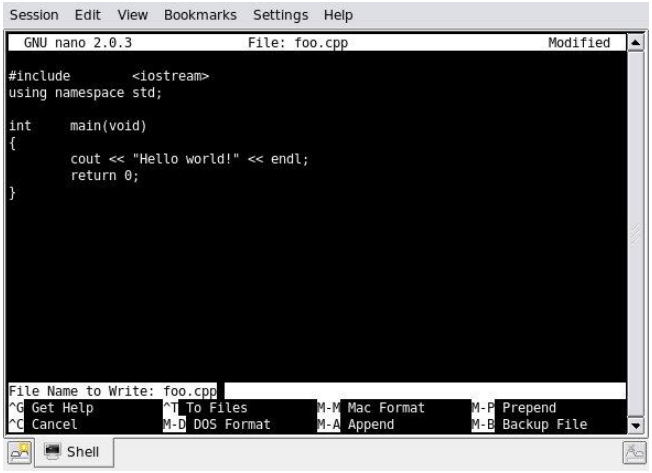


This point you can type in the code for your program:

- Enter the source code exactly as you see it in the window below.
- Notice after your first keystroke, the word “Modified” appears in the upper-right corner; this shows that you've changed the contents of your file but it hasn't been written to the hard drive yet.
- Once you've entered all the code, save the code to the disk file by pressing **Ctrl+o** (look at the next to last row, the second command is **^O** which means to “write out” the file to the hard drive).



After entering the "WriteOut" (Ctrl+o) command, nano will display a prompt on the status line to verify that you really want to write the file contents to the hard drive (see image below):



Go ahead and press the Enter key, and nano will tell you how many lines of text it wrote to disk on the status line. Notice also that the "Modified" indicator in the upper-right corner has disappeared because the file has been saved (see image below):

```
Session  Edit  View  Bookmarks  Settings  Help
  GNU nano 2.0.3                File: foo.cpp

#include        <iostream>
using namespace std;

int     main(void)
{
        cout << "Hello world!" << endl;
        return 0;
}

                          [ Wrote 8 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
Shell
```

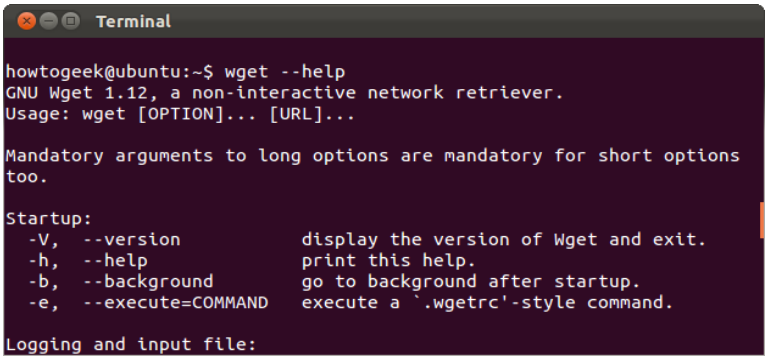
At this point you could exit the nano program (Ctrl+x) to go back to the shell prompt and compile your program with g++. Or, if you'd like to read more about what's available in nano, there's online help that's always available when you're running nano, just type Ctrl+g, which brings you to this screen:

```
Session  Edit  View  Bookmarks  Settings  Help
  GNU nano 2.0.3                File: foo.cpp

Main nano help text

 The nano editor is designed to emulate the functionality and ease-of-use of
 the UW Pico text editor.  There are four main sections of the editor.  The top
 line shows the program version, the current filename being edited, and whether
 or not the file has been modified.  Next is the main editor window showing the
 file being edited.  The status line is the third line from the bottom and
 shows important messages.  The bottom two lines show the most commonly used
 shortcuts in the editor.

 The notation for shortcuts is as follows: Control-key sequences are notated
 with a caret (^) symbol and can be entered either by using the Control (Ctrl)
 key or pressing the Escape (Esc) key twice.  Escape-key sequences are notated
 with the Meta (M-) symbol and can be entered using either the Esc, Alt, or
 Meta key depending on your keyboard setup.  Also, pressing Esc twice and then
 typing a three-digit decimal number from 000 to 255 will enter the character
 with the corresponding value.  The following keystrokes are available in the
 main editor window.  Alternative keys are shown in parentheses:

^L Refresh          ^Y Prev Page        ^P Prev Line        M-\ First Line
^X Exit             ^V Next Page        ^N Next Line        M-/ Last Line
Shell
```

3.10 Working with the help (man).

-h or –help

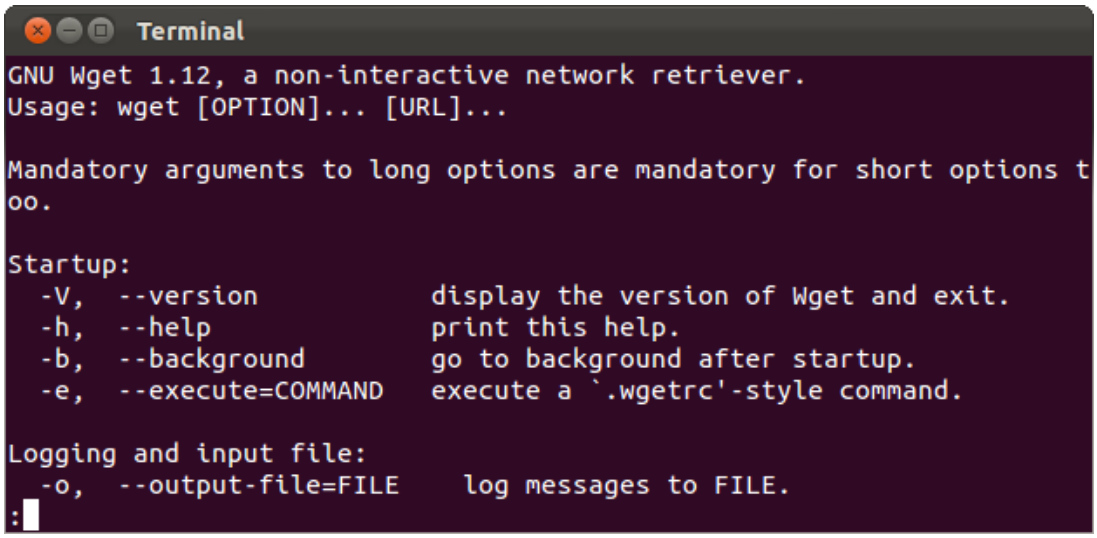
If you're not sure how to use a specific command, run the command with the **-h** or **–help**switches. You'll see usage information and a list of options you can use with the command. For example, if you want to know how to use



the **wget** command, type **wget –help** or **wget -h**.

This will often print a lot of information to the terminal, which can be inconvenient to scroll through. To read the output more easily, you can pipe it through the **less** command, which allows you to scroll through it with the arrow keys on your keyboard. For example, use the following command to pipe wget's help output through less:

wget –help | less



Press **q** to close the less utility when you're done.

To find a specific option, you can pipe the output through the **grep** command. For example, use the following command to search for options that contain the word "proxy":

wget –help | grep proxy

```
howtogeek@ubuntu:~$ wget --help | grep proxy
       --no-proxy                 explicitly turn off proxy.
       --proxy-user=USER        set USER as proxy username.
       --proxy-password=PASS    set PASS as proxy password.
howtogeek@ubuntu:~$
```

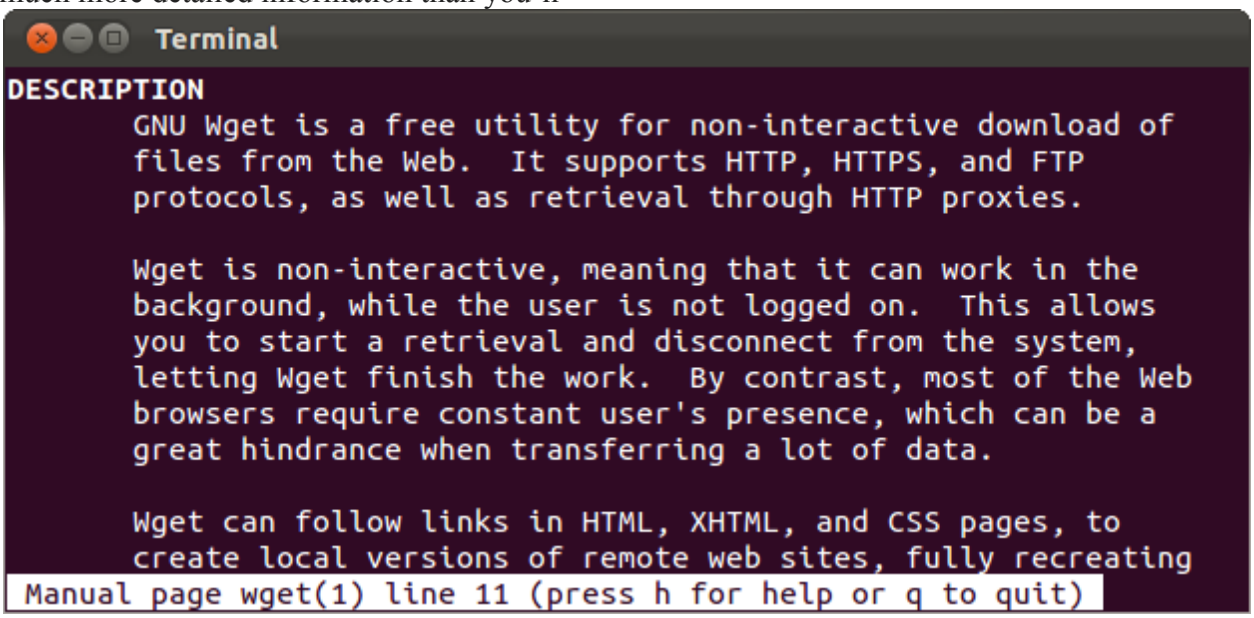
help

The **help** command shows a short list of the commands built into the Bash shell itself.

```
howtogeek@ubuntu:~$ help
GNU bash, version 4.2.10(1)-release (i686-pc-linux-gnu)
These shell commands are defined internally.  Type `help' to see this
 list.
Type `help name' to find out more about the function `name'.
Use `info bash' to find out more about the shell in general.
Use `man -k' or `info' to find out more about commands not in this li
st.

A star (*) next to a name means that the command is disabled.

 job_spec [&]                     history [-c] [-d offset] [n] o>
 (( expression ))                 if COMMANDS; then COMMANDS; [ >
 . filename [arguments]           jobs [-lnprs] [jobspec ...] or>
```

Man

The **man** command shows detailed manuals for each command. These are referred to as "man pages." For example, if you wanted to view the man page for the **wget** command, you'd type**man wget**. Man pages generally contain much more detailed information than you'll



get with the **-h** or **–help** options

Type **man intro** to see a detailed introduction to using the shell on Linux.

```
Terminal
       login(1).   The  program  login  now starts a shell (command
       interpreter) for you.  In case of a graphical login, you get
       a  screen with menus or icons and a mouse click will start a
       shell in a window.  See also xterm(1).

   The shell
       One types commands to the shell,  the  command  interpreter.
       It is not built-in, but is just a program and you can change
       your shell.  Everybody has her own favorite one.  The  stan-
       dard  one  is  called sh.  See also ash(1), bash(1), csh(1),
       zsh(1), chsh(1).

       A session might go like

 Manual page intro(1) line 41 (press h for help or q to quit)
```

```
Terminal
NOTES
       Linux is a flavor of Unix, and as a first approximation  all
       user commands under Unix work precisely the same under Linux
       (and FreeBSD and lots of other Unix-like systems).

       Under Linux there  are  GUIs  (graphical  user  interfaces),
       where  you  can  point and click and drag, and hopefully get
       work done without first reading lots of documentation.   The
       traditional  Unix  environment is a CLI (command line inter-
       face), where you type commands to tell the computer what  to
       do.   That is faster and more powerful, but requires finding
       out what the commands are.  Below a  bare  minimum,  to  get
       started.

 Manual page intro(1) line 24 (press h for help or q to quit)
```

To search a man page, type a /, followed by your query, and press Enter. For example, to search a man page for the word shell, type **/shell** while reading the man page and press Enter.

Check your Progress 3

1. What is Shell?

A._____

_____

_____

2. Which command is used to display the list of content in a folder?

A._____

3. Which command is used to clear the screen?

A._____

4. Which command is used to display the contents of a file?

A._____

5. Which command is used to creates a new directory?

A._____

6. Which command is used to display the current system date?

A._____

7. Which command is used to shut down the Linux environment ?

A._____

3.11 Let us sum up

Linux is a multi-user multi-processing operating system which comes in 2 versions desktop and server versions. Linux is free and highly secure operating system. Using this operating system keeps viruses and malware away. Linux was invented by ***Linus Torvalds***. Linux when installed is installed in a un partitioned disk since during installation it creates 3 partitions .These are termed as root, boot and swap. The root is the top level directory and it defined by the symbol /. Under root there are a number of directories which are designed for a particular purpose. The boot comprises of OS kernel along with files used during boot strap process. The swap is used a virtual memory which stores data for processing when there is less space in RAM. Root takes 5GB of hard disk. The swap takes twice of the memory space taken by RAM. If RAM takes 2 GB of space the swap takes 4GB of space. The boot takes 200 MB of space. Users are of 5 types namely super, system, normal, network and Pseudo. The root home directory identified by ~ symbol is used by the super user to create a folders and files. The home directory is used by the other users to store their folders and files. The super user has all the rights over the system. users can be created and passwords can be allotted to them. This enables tight security in the system. The kernel is the nucleus of the operating system. commands given via keyboard are taken by the shell and sent to kernel for execution. The kernel executes the command and sends it to the shell which gives the result to the user.

Commands that are executed at the command line are termed as command line commands. The commands that are used to manage the files ,folders and administer the Linux system are termed as File management commands. Nano is used as an editor which is used in Linux .XFree86 is free version of Unix X windows system.it provides graphical interface for users. It uses a client server model which means that the main operating system works at one machine

And the rest of the systems communicate to it through sessions. By this client systems that are available locally can communicate with the server and those that are very far away or remote systems communicate with the server. This enables data security and high speed transmission of data. Data is located at one place which is safe and secure. Sawfish is a window manager for the X Window System. It aims to manage windows in the most flexible and attractive manner possible. Gnome is a Windows-like desktop system that works on UNIX and Linux. Working with a command if help is needed

regarding the command **command -h** gives the help relating to the command. The man command also used for the same purpose man command name gives the help relating to the command.

3.12 Key Words

1. Partition :- A **partition** is a logical division of a **hard disk** created so that you can have different operating systems on the same **hard disk** or to create the appearance of having separate **hard drives** for file management, multiple users, or other purposes.
2. GB :- Gigabyte = 1024 MB
3. MB :- Mega Byte = 1024 KB
4. KB :- Kilo Byte =1024 bytes
5. RAM :- Random access memory .

3.13 References

1. Linux Bible by Christopher Negus
2. Google.com

3.14 Answers to check your Progress

Answers to check your Progress 1

A.1. The kernel is the essential center of a computer operating system, the core that provides basic services for all other parts of the operating system. A synonym is nucleus. A kernel can be contrasted with a shell, the outermost part of an operating system that interacts with user commands.

Answers to check your Progress 2

A.1. Gnome is a desktop environment that is developed for the Linux operation system. Gnome helps in creating new files, folders, copying files from one folder to the other, Delete files, rename files. Because of the graphical user interface a user does not know the commands on the command line can effortlessly manage with files and folders.

A.2. XFree86 is free version of Unix X windows system.it provides graphical interface for users. It is uses a client server model which means that the main operating system works at one machine and the rest of the systems communicate to it through sessions. By this client systems that are available locally can communicate with the server and those that are very far away or remote systems communicate with the server. This enables data security and high speed transmission of data. Data is located at one place which is safe and secure. The X free 86 graphical interface enables ease of access such a file creation, modification and other features for the authorized user of the system. It comes in two versions 3x and 4x.

A.3. Command line commands are the commands that are issued at the terminal mode of the Linux. Unlike the graphical user interface these commands given on the terminal mode. Commands given at the terminal mode are taken care by the shell that takes keyboard commands and passes them to the operating system to carry out.

Answers to c heck your Progress 3

A.1 The shell is a program that takes keyboard commands and passes them to the operating system to carry out.

A.2 ls command

A.3 clear

A.4 cat filename

A.5 mkdir directoryname

A.6 date

A.7 halt

Unit -4
Linux Part-II

Learning objectives

After the Completion of this unit you should be able to

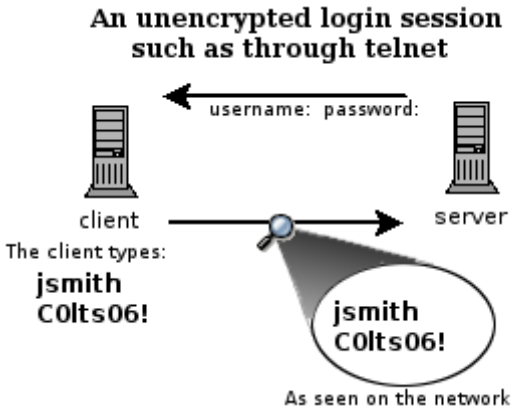
1. Adding Users and Group
2. Changing User Password
3. Working the command SU and SUDO.
4. How to installed of YUM , YAST & RPM
5. How to Install Webmin.

Structure

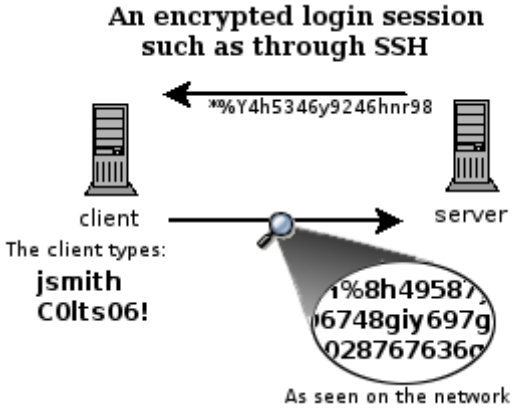
4.1 SSH and X-Forwarding

What Is SSH?

There are a couple of ways that you can access a shell (command line) remotely on most Linux/Unix systems. One of the older ways is to use the telnet program, which is available on most network capable operating systems. Accessing a shell account through the telnet method though poses a danger in that everything that you send or receive over that telnet session is visible in plain text on your local network, and the local network of the machine you are connecting to. So anyone who can "sniff" the connection in-between can see your username, password, email that you read, and commands that you run. For these reasons you need a more sophisticated program than telnet to connect to a remote host.



SSH, which is an acronym for Secure Shell, was designed and created to provide the best security when accessing another computer remotely. Not only does it encrypt the session, it also provides better authentication facilities, as well as features like secure file transfer, X session forwarding, port forwarding and more so that you can increase the security of other protocols. It can use different forms of encryption ranging anywhere from 512 bit on up to as high as 32768 bits and includes ciphers like AES (Advanced Encryption Scheme), Triple DES, Blowfish, CAST128 or Arcfour. Of course, the higher the bits, the longer it will take to generate and use keys as well as the longer it will take to pass data over the connection.



These two diagrams on the left show how a telnet session can be viewed by anyone on the network by using a sniffing program like Ethereal (now called Wireshark) or tcpdump. It is really rather trivial to do this and so anyone on the network can steal your passwords and other information. The first diagram shows user jsmith logging in to a remote server through a telnet connection. He types his username jsmith and password C0lts06!, which are viewable by anyone who is using the same networks that he is using.

The second diagram shows how the data in an encrypted connection like SSH is encrypted on the network and so cannot be read by anyone who doesn't have the session-negotiated keys, which is just a fancy way of saying the data is scrambled. The server still can read the information, but only after negotiating the encrypted session with

This tutorial isn't going to cover how to install SSH, but will cover how to use it for a variety of tasks. Consult your Linux distribution's document for information on how to setup OpenSSH.

Chances are that if you are using a version of Linux that was released after 2002, that you already have OpenSSH installed. The version of SSH that you will want to use on Linux is called OpenSSH. As of this writing (October 2009), the latest version available is 5.3, but you may encounter versions from 3.6 on up. If you are using anything lower than version 3.9, I'd strongly advise you to upgrade it.

X-Forwarding.

Now that you've seen general TCP port forwarding, we move to a new topic: forwarding of X protocol connections. X is a popular window system for Unix workstations, and one of its best features is its transparency. Using X, you can run remote X applications that open their windows on your local display (and

vice versa, running local applications on remote displays). Unfortunately, the inter-machine communication is insecure and wide open to snoopers. But there's good news: SSH *X forwarding* makes the communication secure by tunneling the X protocol.

X forwarding also addresses some firewall-related difficulties. Suppose you're a system administrator with a set of exposed production machines on the other side of a firewall from you. You log into one of these machines using SSH, and want to run an graphical performance-monitoring tool, such as Solaris's perfmon, that uses the X Window System. You can't, though, because to do that, the external machine needs to make a TCP connection back to the internal machine you started on, and the firewall blocks it (as it should, since X is quite insecure). X forwarding solves this problem, permitting X protocol connections to pass through the firewall, securely tunneled via SSH.

How X Forwarding Works

Although X clients can communicate with remote X servers, this communication isn't secure. All interactions between the X client and server, such as keystrokes and displayed text, can be easily monitored by network snooping because the connection isn't encrypted. In addition, most X environments use primitive authentication methods for connecting to a remote display. A knowledgeable attacker can get a connection to your display, monitor your keystrokes, and control other programs you're running.

Once again, SSH comes to the rescue. An X protocol connection can be routed through an SSH connection to provide security and stronger authentication. This feature is called X forwarding.

4.2 Managing compressed archives with zip and tar

Data compression has been extremely useful to us over the years. Whether its a zip file containing images to be sent in a mail or a compressed data backup stored on a server, we use data compression to save valuable hard drive space or to make the downloading of files easier. There are compression formats out there which allow us to sometimes compress our data by 60% or more. I'll run you through using some of these formats to compress and decompress files and directories on a Linux machine. We'll cover the basic usage of zip, tar, tar.gz and the tar.bz2 formats. These are some of the most popular formats for compression used on Linux machines.

ZIP

Zip is probably the most commonly used archiving format out there today. Its biggest advantage is the fact that it is available on all operating system platforms such as Linux, Windows, and Mac OS, and generally supported out of the box. The downside of the zip format is that it does not offer the best level of compression. Tar.gz and tar.bz2 are far superior in that respect. Let's move on to usage now. To compress a directory with zip do the following:

zip -r archive_name.zip directory_to_compress

Here's how you extract a zip archive:

unzip archive_name.zip

TAR

Tar is a very commonly used archiving format on Linux systems. The advantage with tar is that it consumes very little time and CPU to compress files, but the compression isn't very much either. Tar is probably the Linux/UNIX version of zip – quick and dirty. Here's how you compress a directory:

tar -cvf archive_name.tar directory_to_compress

And to extract the archive:

tar -xvf archive_name.tar.gz

This will extract the files in the archive_name.tar archive in the current directory. Like with the tar format you can optionally extract the files to a different directory:

tar -xvf archive_name.tar -C /tmp/extract_here/

TAR.GZ

This format is my weapon of choice for most compression. It gives very good compression while not utilizing too much of the CPU while it is compressing the data. To compress a directory use the following syntax:

tar -zcvf archive_name.tar.gz directory_to_compress

To decompress an archive use the following syntax:

tar -zxvf archive_name.tar.gz

This will extract the files in the archive_name.tar.gz archive in the current directory. Like with the tar format you can optionally extract the files to a different directory:

 # tar -zxvf archive_name.tar.gz -C /tmp/extract_here/

TAR.BZ2

This format has the best level of compression among all of the formats I've mentioned here. But this comes at a cost – in time and in CPU. Here's how you compress a directory using tar.bz2:

 # tar -jcvf archive_name.tar.bz2 directory_to_compress

This will extract the files in the archive_name.tar.bz2 archive in the current directory. To extract the files to a different directory use:

 # tar -jxvf archive_name.tar.bz2 -C /tmp/extract_here/

Data compression is very handy particularly for backups. So if you have a shell script that takes a backup of your files on a regular basis you should think about using one of the compression formats you learned about here to shrink your backup size.

Over time you will realize that there is a trade-off between the level of compression and the time and CPU taken to compress. You will learn to judge where you need a quick but less effective compression, and when you need the compression to be of a high level and you can afford to wait a little while longer.

Check your progress 1

Q.1 What is SSH?

A._____

_____

_____

Q.2 What is X-forwarding?

A_____

_____

_____

Q.3 How to compress a file?

A_____

_____

4.3 Working with the GNU screen

GNU Screen is a tool which works with a terminal session to allow users to resume a session after they have disconnected. Screen prevents a session from "timing out" or disconnecting SSH connections or local terminal emulators. A single Screen session has the ability to host multiple sessions or "windows." Screen may be used for a variety of tasks such as maintaining persistent IRC sessions and multitasking in a terminal environment.

Installing GNU Screen

The section covers installing Screen on a number of different systems. Examples have been provided to simplify the installation process.

When installing Screen you will need root privileges. The examples provided do not use the root account. If you are using your root login then the sudo before the commands is not necessary.

For a Debian or Ubuntu system use the following commands to update, upgrade, and install Screen:

```
sudo apt-get update

sudo apt-get upgrade

sudo apt-get install screen
```

For an Arch Linux system, the following commands are used to update and install Screen:

```
sudo pacman -Sy
sudo pacman -S screen
```

For a CentOS or Fedora system use the following commands to update the system and install Screen:

```
sudo yum update
sudo yum install screen
```

Red hat Linux has screen on /usr/bin/screen

To see if screen is the path use command

```
[root@office ~]# which screen
/usr/bin/screen
```

If you do not have screen installed install it using the yum command

```
[root@office ~]# yum install screen
```

To start the screen use the command

```
[root@office ~]# screen
```

To see the screen help page use the command

[root@office ~]# **"Ctrl-a"** then **"?"**.

To create a new window use the command "Ctrl – a" "c"

To switch between windows type the command

"Ctrl –a" "n" – To go to next window

"Ctrl –a" "p" – To go to previous window

To detach from a screen use the command

"Ctrl –a" "d"

To reattach to a screen use the command

```
[jeffh@office ~]$ screen-r
```

However, if you have multiple screens you may get this:

```
[jeffh@office ~]$ screen-r
There are several suitable screens on:
31917.pts-5.office      (Detached)
31844.pts-0.office      (Detached)
Type "screen [-d] -r [pid.]tty.host"to resume one of them.
```

If you get this, just specify the screen you want.

```
[jeffh@office ~]$ screen-r  31844.pts-0.office
```

When you are done with your work, I recommend you stop the session instead of saving it for later. To stop screen you can usually just type exit from your shell. This will close that screen window. You have to close all screen windows to terminate the session.

You should get a message about screen being terminated once you close all windows.

```
[screenis terminating]
```

Alternatively, you can use **"Ctrl-a" "k"**. You should get a message if you want to kill the screen.

4.4 How to add users and groups

Adduser and addgroup to the system according to command line options and configuration information in /etc/adduser.conf. They are friendlier front ends to the low level tools like useradd, groupadd and usermod programs, by default choosing Debian policy conformant UID and GID values, creating a home directory with skeletal configuration, running a custom script, and otherfeatures. adduser and addgroup can be run in one of five modes:

There are 5 types of users

1. Super user :- Also termed as Root user. Has the privileges to Administer the Linux server. Has the control to limit the access of other users.
   The command to create a super user is
   useradd – u 0 username
   Here userid =0 is set for superuser
2. System user: - This user is created by the Linux Operating System. Have more privileges than the normal user.
   The command to create a system user is
   Useradd – u uid username
   uid is user id which ranges from 1 – 499
3. Normal user: - These users are created by the super user. They can access only those privileges given by the Super user.
   The command to create a normal user is
   useradd username.
   The number of users that can be added in 32 bit computer is 2^{16}
   The number of users that can be added to 64 bit computer is 2^{32}
4. Network user: - Users who opt this type of user accounts are network engineers and system administrators who monitor the network activity. Network users can be created at the server level and creation of network user is beyond the scope of the book.
5. Pseudo user: - This is a replica of the Super user which is granted by the super user to user accounts.
   The command to create a Pseudo user is
   usermod -aG sudo username

4.5 Listing the users

cat /etc/passwd

4.6 Creating A new user

adduser username

4.7 Adding password to a user

#passwd username :- this command would ask the user to enter the new password as shown below

#new password

And confirm password as shown below

#confirm password

4.8 Changing the password of a user

#passwd username

```
[root@localhost Desktop]# passwd gopal
Changing password for user gopal.
New password:
```

```
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost Desktop]#
```

4.9 Deleting A User

#userdel username

```
[root@localhost Desktop]# userdel ram
```

#userdel-r username

4.10 Locking A User

usermod –l username

4.11 Unlocking A User

usermod –u username

4.12 Rename A User

usermod –l newusername oldusername

4.13 User and group management

Users and groups is the core element of a Red Hat enterprise Linux System Admin (RHCL).
Users and groups are used to control access to file and resources. Users all that are created belong to the primary group.
The user can be allocated to secondary groups such as hrgroup, reception in offices. A user that is created can be added to one or more groups.

4.14 Creating a group

#groupadd groupname

Example

#groupadd admin

4.15 Adding A Single User To A Group

#usermod –G groupname username

4.16 Removing A Single User From A Group

#gpasswd –a username group

4.17 Adding multiple users to a group

#gpasswd –M user1,user2,user3 groupname

4.18 Listing all users in a group

#grep groupname /etc/group

4.19 Removing Users From A Group

#gpasswd –d username1,username2,groupname

4.20 Renaming A Group

#groupmod –n newgroupname oldgroupname

4.21 Make a user as an administrator

#gpasswd – A username groupname

4.22 List which group the user belongs

groups username

4.23 Delete A Group

groupdel groupname

4.24 Working with Permissions

The permission is applied at 3 levels

1) Owner/user level
2) Group level
3) Other users

Reading -4
Writing – 2
Execute- 1
No permission – 0

Making the file read only for the owner

Check your progress 2

Q.1 What are the types of users?

A._____

Q.2 What is the command to create a super user ?

A._____

Q.3 What is the command to create a system user?

A._____

Q.4 What is the command to create a normal user?

A._____

Q.5 What is the command to create a pseudo user?

A._____

Q.6 What is the command to create a group?

A._____

Q.7 What is the command to add a single user under a group?

A._____

Q.8 What is the command to add multiple users under a group?

A._____

Q.9 What is the command to remove a user from a group?

A._____

Q.10 What is the command to delete a user?

A._____

Q.11 What is the command to rename a user?

A._____

Q.12 What is the command to lock a user?

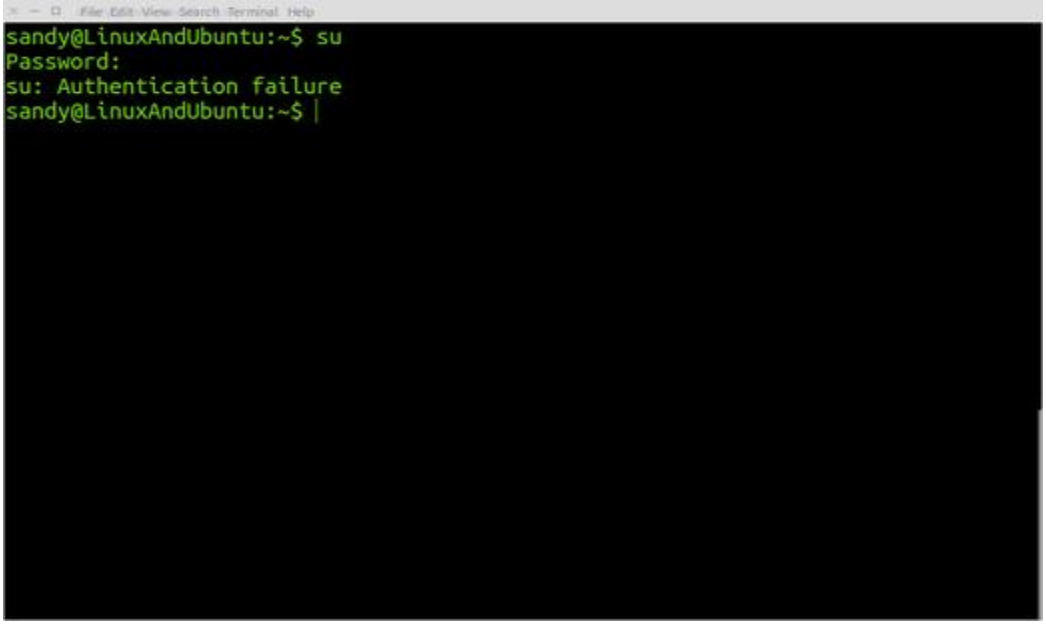
A._____

Q.13 What is the command to unlock a user?

A._____

4.3 Working with su

The Linux command 'su' is used to switch from one account to another. The Linux command **su**, sometimes described as *substitute user*, *super user*, or *switch user*, is used by a computer user to execute commands with the privileges of another user account. When the command is used without specifying the new user id as a command line argument, it defaults to using the superuser account (user id 0) of the system. By default if I type su it means super user.The computer will ask me to enter super user password.



If I type su sandy it means switch to user Sandy as shown below.

shan@localhost:~$ su Sandy
Password:
Sandy@localhost:/home/shan$exit

```
logout
shan@localhost:~$
```

To come out of the user type exit command as shown below.

When used with a hyphen (su -) it can be used to start a login shell. In this mode users can assume the user environment of the target user:

```
Shan@localhost:~$ su - Sandy
Password:
Sandy@localhost:~$
```

SU - change user ID or become superuser

4.4 Working with sudo

The sudo command is used when a normal user does not have the rights of the root user. For example I am a normal user and Ram my system administrator is on leave. I want to install a package on Linux as it is very urgent for the company. That time sudo command comes to help

The command to run the sudo command is **sudo apt –get install wine**

Let me describe what we have written here

- Sudo :- this give the access to the root user access for 15 minutes
- apt –get install :- this command is used to install a software
- wine :- this is the name of the software that we want to install
- sudo su **:-** This command allows the normal user to be a super user.

4.5 Change user Password

passwd username :- This command is used to change the password of the user

4.6 Printing

The printing command in Linux is

```
# cat thesis.txt > /dev/lp
```

4.7 Installing Software with Rpm

RPM stands for Red Hat Package Manager

1. RPM is free and released under GPL (General Public License).
2. RPM keeps the information of all the installed packages under /var/lib/rpm database.
3. RPM is the only way to install packages under Linux systems, if you've installed packages using source code, then rpm won't manage it.
4. RPM deals with .rpm files, which contains the actual information about the packages such as: what it is,from where it comes, dependencies info, version info etc.

RPM is used for 5 purposes

1. Install : It is used to install any RPM package.
2. Remove : It is used to erase, remove or un-install any RPM package.
3. Upgrade : It is used to update the existing RPM package.
4. Verify : It is used to query about different RPM packages.
5. Query : It is used for the verification of any RPM package.

Source : https://www.youtube.com/watch?v=72qV32isteQ

1. Install

Step-1

Insert the Linux dvd in the dvd rom drive

Step-2

cd /~ :- this command is to get into the root home directory

Step-3

[root@localhost ~]mkdir rhce :- this command is used to create a rhce directory in the root home folder.

Step-4

[root@localhost ~] mount /dev/dvd /rhce

/* The above command is used mound a dvd to the media directory */

Step-5

[root@localhost ~] cd /rhce

[root@ localhost rhce] cd /packages

Step-6

/# ls command to list all rpm packages

Step-7

[root@localhost Packages] rpm –ivh package name

i= install

v =verbose

h=hash

Uninstall an RPM package

[root@localhost Packages] rpm – e packagename

Upgrade an RPM package

[root@localhost Packages] rpm – U packagename

To query every rpm command installed in the system

rpm –qa

To query every file in an installed package

[root@localhost Packages] Rpm –ql packagename

Example :-

[root@localhost Packages] Rpm –ql BitTorrent

To verify an RPM package

[root@localhost Packages] # rpm – Vp packagename

Example :-

[root@localhost Packages] Rpm –Vp BitTorrent

To verify all RPM packages

[root@localhost Packages] # rpm – Va

4.8 Working with Yum

4.8.1 Yum configuration

Source :- https://www.youtube.com/watch?v=ainF0UqbdOQ

Step-1

Insert the Linux dvd in the dvd rom drive

Step-2

cd /~ :- this command is to get into the root home directory

step-3

[root@server1 ~]mkdir media :- this command is used to create a media directory in theroot home folder.

step-4

[root@server1 ~] mount /dev/cdrom /media

/* The above command is used mound a dvd to the media directory */

Step-5

[root@server1 ~] cd /media/packages

/* The above command is used to get into the packages folder */

step-6

[root@server1 Packages] ls

The above command is used to list all the rpm packages in the package directory

Step-7

We have to install 3 RPM packages. These are

1. **vsftpd***
2. **deltarpm***
3. **python-deltarpm***

<u>commands to install these 3 packages are</u>

[root@server1 Packages] rpm –ivh vsftpd*
[root@server1 Packages] rpm –ivh deltarpm*
[root@server1 Packages] rpm –ivh python-deltarpm*

Step-8

Once these three are done we can install the package createrepo

[root@server1 Packages] rpm –ivh createrepo*

This command is basically for creating repositories. Without this package we cannot create a

repository

Step-9

Now get back to the root directory

[root@server1 Packages] cd /

Step-10

[root@server1 /]#

Here we will create a folder named Repo by using the mkdir command

[root@server1 /]# mkdir Repo

Step -11

Move back to the media folder where we have cd rom loaded

[root@server1 /]# cd /media

step-12

Now we have to get certain files from here

Step-13

[root@server1 media]# cp –rvf /Packages/RPM –GPG-KEY-redhat-release /Repo

Step-14

[root@server1 media]# createrepo –database /Repo/Packages/

Step-15

[root@server1 media]# ls /Repo

/*** this option gives the keys **********/**

Packages:RPM-GPG-KEY-redhat_release

Step-16

Go to the root and create a repository file

[root@server1 media]# cd /

[root@server1 /]#

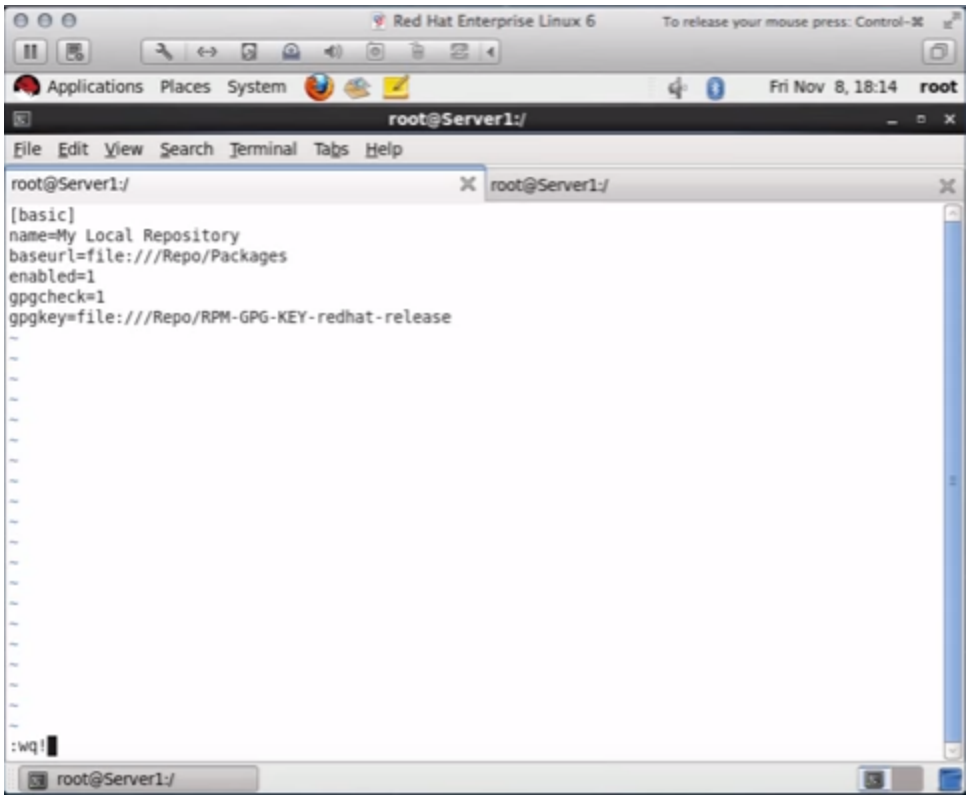
Step-17

Now we create a repository file as shown below

[root@server1 /]# vim etc/yum.repos.d/basic.repo

Step-18

Going to the insert mode of vim by pressing i



Step-19

/*******Now to check whether all the configurations are correctly done or not ******/**

[root@server1 /]# yum repolist

4.8.2 Install a package using Yum

Step-1

/********how to install a package using yum **********/**

[root@server1 /]# yum –y install kdebase*

/********/**

4.8.3 How to uninstall a package using Yum

Step-1

[root@server1 /]# yum remove kdebase

4.8.4 How to update a package using Yum

Step-1

[root@server1 /]# yum update kdebase

4.8.5 Listing all packages using Yum

[root@server1 /]# yum list | less

4.8.6 List available group packages using yum

In Linux, number of packages are bundled to particular group. Instead of installing individual packages with yum, you can install particular group that will install all the related packages that belongs to the group. For example to list all the available groups, just issue following command. **[root@server1 /]#**

yum grouplist

4.8.7 Install a group package using YUM

To install a particular package group, we use option as groupinstall. For example, to install "MySQL Database", just execute the below command.

[root@server1 /]# yum groupinstall 'MySQL Database'

4.8.8 Update a group package using YUM

To update any existing installed group packages, just run the following command as shown below.

[root@server1 /]# yum groupupdate 'MySQL Database'

4.8.9 Remove a group package using YUM

To remove a group package

[root@server1 /]# yum groupremove 'MySQL Database'

4.8.10 List a package using YUM

[root@server1 /]# yum list packagename

[root@server1 /]# yum list openssh

4.8.11 Search for a package using YUM

[root@server1 /]# yum search packagename

[root@server1 /]# yum search vsftpd

4.8.12 Get information about a package using YUM

[root@server1 /]# yum info firefox

4.8.13 Check updates using YUM

[root@server1 /]# yum check-update

Check your progress 3

Q.1 What is su

A.1_____

Q.2 What is sudo

A.2_____

Q.3 What is the command to print a file

A.3_____

Q.4 What is RPM and YUM

A.4_____

4.9 Yast

Yast is a graphical user interface to which can do the following jobs :

- Install and remove software
- Set up your printer
- Configure the firewall
- Enable and disable system services
- Configure network sharing (samba)
- Format and partition your drives
- Enable NTP daemon
- And much, much more...

But since we are confined to installation of software using Yast we shall be discussing about Yast installation.

4.9.1 Working with Yast

Step-1

Step-2

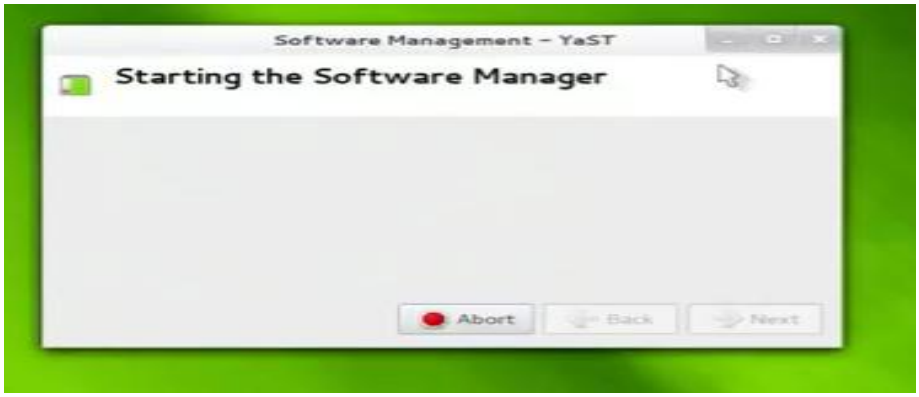
Step-3


Activities
Fri 15 Feb, 00:04
Mantis
Windows
Applications
inst
Install/Remove S...
Software Update
GNOME Characte...
American Mustac...
H.I. Hardware Inf...
Installing Apache...
How to Install a...
ICEfaces (Ajax F...
Instant Grammar...
Links to WEB Sit...
Wikipedia
Google

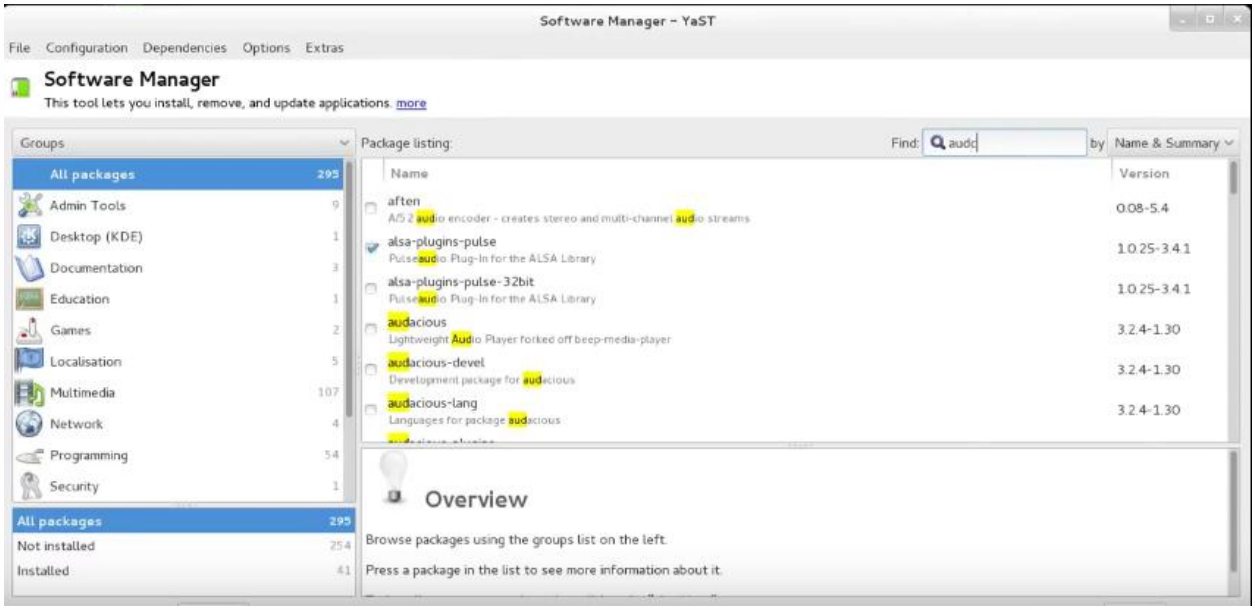
Step-4


Password Needed
The requested action needs further authentication
Please enter the root password to continue
Password:
Details
Cancel
Continue

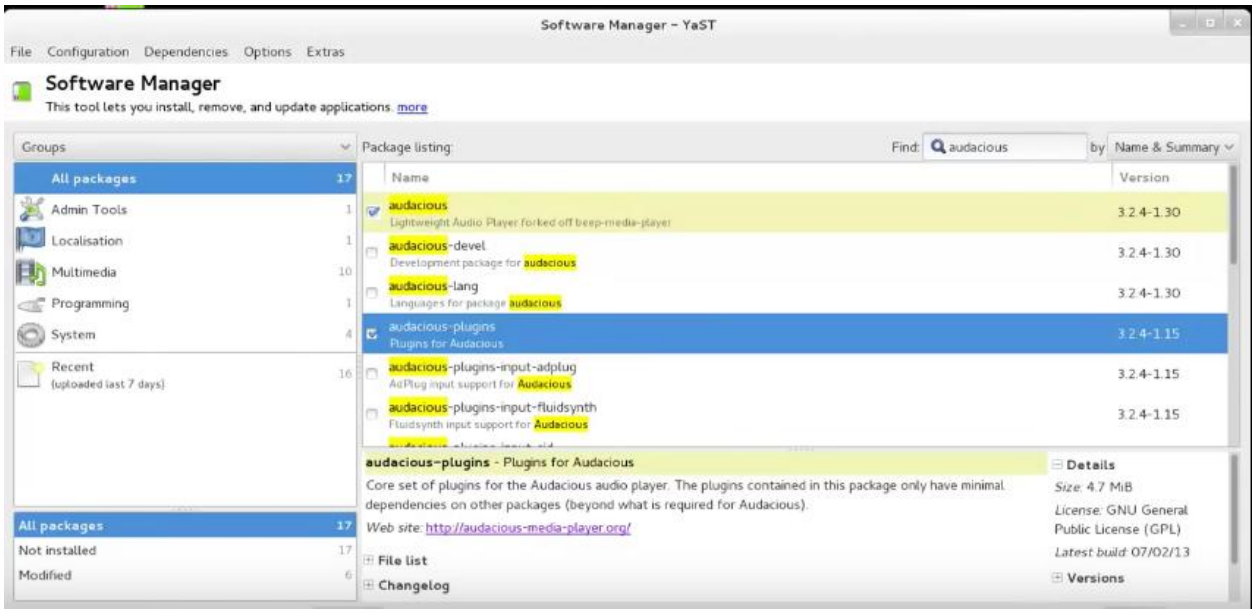
Step-5


Software Management – YaST
Starting the Software Manager
Abort
Back
Next

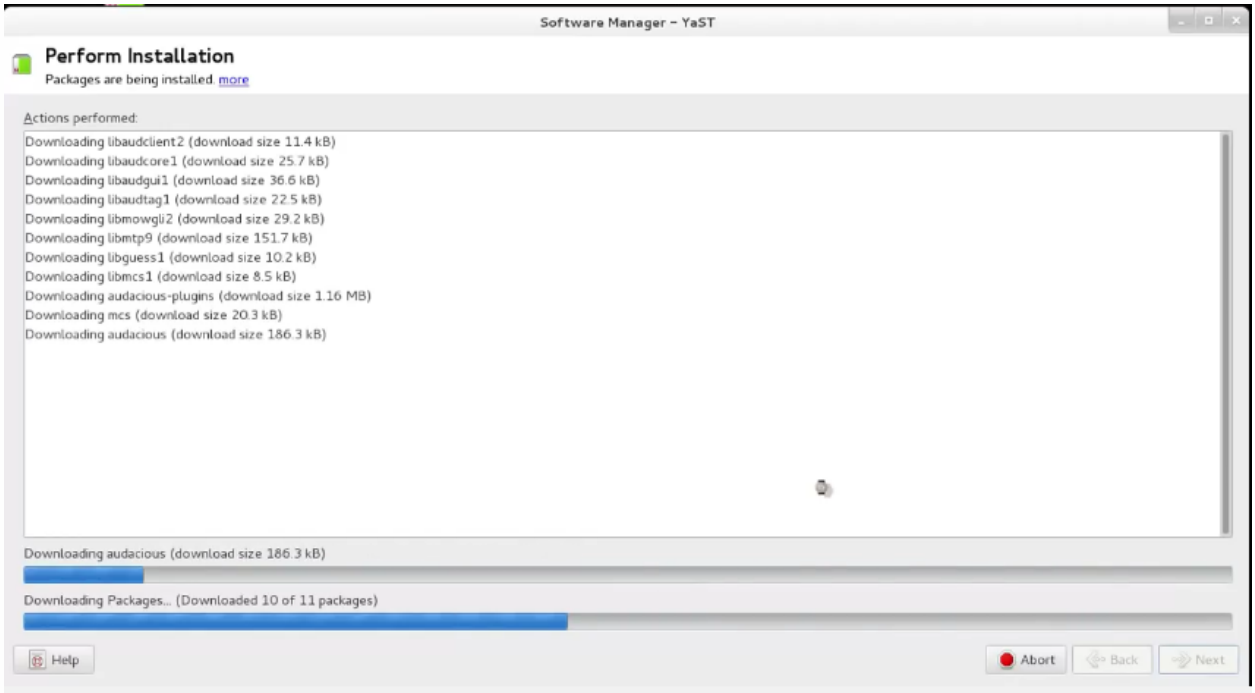
Step-6


Software Manager – YaST
File Configuration Dependencies Options Extras
Software Manager
This tool lets you install, remove, and update applications. more
Groups
Package listing:
Find: audc
by Name & Summary
All packages 295
Admin Tools 9
Desktop (KDE) 1
Documentation 3
Education 1
Games 2
Localisation 5
Multimedia 107
Network 4
Programming 54
Security 1
All packages 295
Not installed 254
Installed 41
Name
Version
aften
0.08-5.4
alsa-plugins-pulse
1.0.25-3.4.1
alsa-plugins-pulse-32bit
1.0.25-3.4.1
audacious
3.2.4-1.30
audacious-devel
3.2.4-1.30
audacious-lang
3.2.4-1.30
Overview
Browse packages using the groups list on the left.
Press a package in the list to see more information about it.

Step-7


Software Manager - YaST
File Configuration Dependencies Options Extras
Software Manager
This tool lets you install, remove, and update applications. more
Groups
Package listing:
Find: audacious
by: Name & Summary
All packages 17
Admin Tools 1
Localisation 1
Multimedia 10
Programming 1
System 4
Recent (uploaded last 7 days) 16
All packages 17
Not installed 17
Modified 6
Name
Version
audacious
Lightweight Audio Player forked off beep-media-player
3.2.4-1.30
audacious-devel
Development package for audacious
3.2.4-1.30
audacious-lang
Languages for package audacious
3.2.4-1.30
audacious-plugins
Plugins for Audacious
3.2.4-1.15
audacious-plugins-input-adplug
AdPlug input support for Audacious
3.2.4-1.15
audacious-plugins-input-fluidsynth
Fluidsynth input support for Audacious
3.2.4-1.15
audacious-plugins - Plugins for Audacious
Core set of plugins for the Audacious audio player. The plugins contained in this package only have minimal dependencies on other packages (beyond what is required for Audacious).
Web site: http://audacious-media-player.org/
File list
Changelog
Details
Size: 4.7 MiB
License: GNU General Public License (GPL)
Latest build: 07/02/13
Versions

Step-8


Software Manager - YaST
Perform Installation
Packages are being installed. more
Actions performed:
Downloading libaudclient2 (download size 11.4 kB)
Downloading libaudcore1 (download size 25.7 kB)
Downloading libaudgui1 (download size 36.6 kB)
Downloading libaudtag1 (download size 22.5 kB)
Downloading libmowgli2 (download size 29.2 kB)
Downloading libmtp9 (download size 151.7 kB)
Downloading libguess1 (download size 10.2 kB)
Downloading libmcs1 (download size 8.5 kB)
Downloading audacious-plugins (download size 1.16 MB)
Downloading mcs (download size 20.3 kB)
Downloading audacious (download size 186.3 kB)
Downloading audacious (download size 186.3 kB)
Downloading Packages... (Downloaded 10 of 11 packages)
Help
Abort
Back
Next

4.10 Webmin

Webmin is a software which is used for

- Managing the server remotely from a web interface
- Managing users and groups.
- Managing soft wares and updating them.
- Monitor servers.
- Schedule backups.
- Manage services.
- Manage networking systems.
- Set cron jobs and many more.

4.10.1 Installing webmin on Ubentu server 10.0

Source :- https://youtu.be/eO3zOVTNF0I

Step-1

Login as super user and enter your password

Step-2

Login as the root

root@test:/# wget http://webmin.com/download/deb/webmin-current.deb

This is going to the webmin website and download the latest version of webmin.

Step-3

Now the latest version of the webmin got downloaded

To check whether the webmin got downloaded or not type **ls command**

This will show a list of files and among them is webmin shown in red colour

webmin_1.550_all.deb

Step-4

In this step we will install webmin

root@test:/# dpkg - I **webmin_1.550_all.deb**

we cannot install because of many dependencies and so we issue another command

root@test:/# apt -get –f install

Step-5

keep pressing Y when it asks for confirmation

Step-6

Webmin gets installed and is on the port 10000

Step-7

Now we need to find the ip address and the command to get the ip address of the server is **ifconfig**. The ip address is **10.1.50.32**

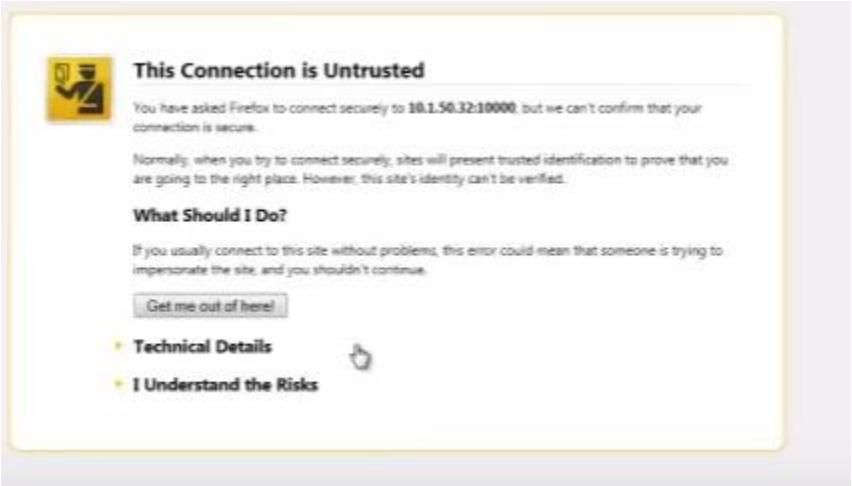
Step-8

Now open the web browser

http://localhost:10000/

Step-9

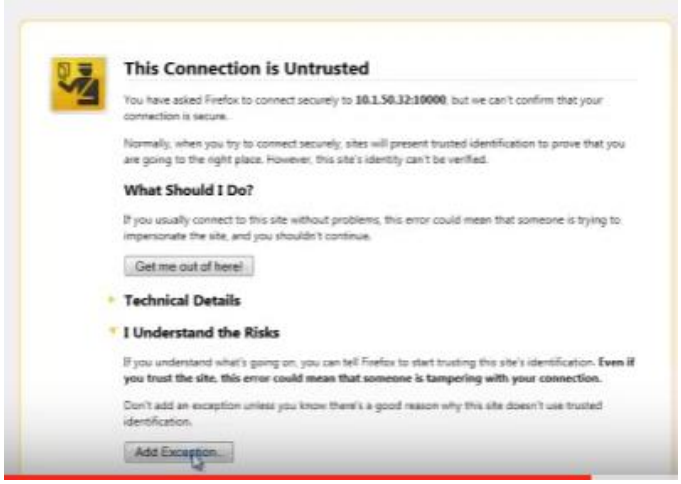
In the beginning it gives a security certificate error



Step-10

Click on I understand the risks

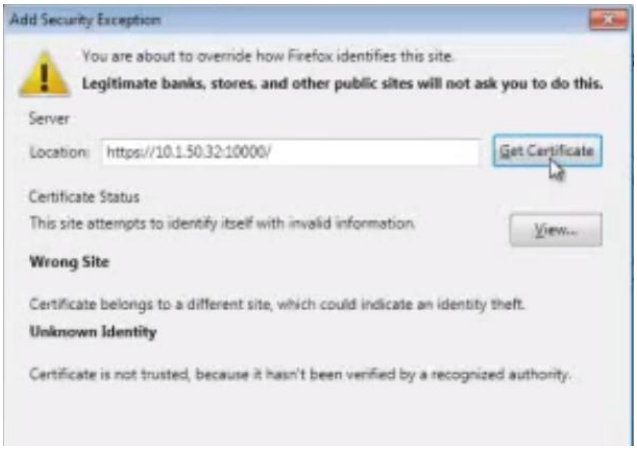
Step-11



Click on Add Exception

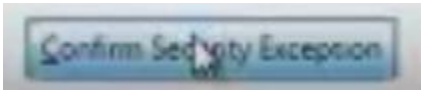
Step-12

Click on Get Certificate



Step-13

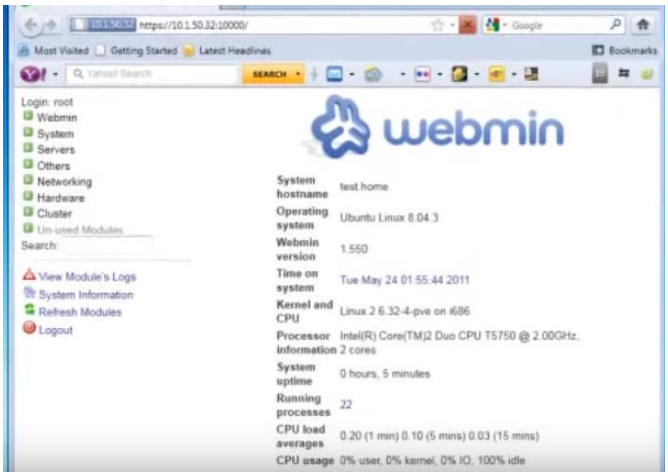
Click on Confirm Security Exception as shown below



Step-14

Step-15

Login as the root and type the password and you can find web min installed



Check your progress 4

Q.1 what is Yast?

A._____

Q.2 What is Webmin?

A._____

4.11 Let us sum up

Summing up SSH is a secure way to access remote machines and machines on the local area network in Linux and UNIX.X-forwarding is used to run x remote applications. SSH is used in X forwarding for secure connection. There are various ways of compressing folders and files. Compressing of folders and files ease transport and space reduction. Compression is done in 4 ways. They are zip,tar,tar,gz,tar.bz2.GNU screen helps us to manage multiple sessions. Users are

of 5 types. They are super user, system users, network users, normal user, pseudo user. Creating the users and groups. Users can be renamed, locked and unlocked . Group is a set of users. Groups are created to group users as one. Groups are created so that we can identify which user belongs to which group. Groups can be created and users can be added and removed from the group. Files those that are created by the user are given permission. The permissions are read, write and execute. These permissions can be issued at owner level,group level and other user level. su and how su is used to switch user and a plain su directs us to super user login.sudo which allows us to do the jobs of super users being a normal user such as installing packages for 15 minutes. sudo followed by su gives the normal user to get into the super user access. Users that are created have a password. The password of the user can be changed by the user. Printing can be done from Linux by issuing commands. Packages can be installed into Linux using RPM,YUM,Yast,Webmin.

4.12 References

1. Linux Bible
2. Google.com

4.13 Check your progress – possible answers

Answers to check your progress 1

A.1 SSH stands for secured shell. This is used to connect to a remote server through telnet in as secured manner so that other cannot llok into the data that is being transmitted. Protects a authenticated user from preying eyes who have bad intentions such as stealing user name and passwords from the network by encryption, transmission and decryption.

A.2 X-forwarding is a feature of UNIX where in multiple X applications located at remote machines can be run by the user.But during access with these machines leads to transparency in the network. Data transmitted in this manner is viable to hacking by a third person. To prevent this kind of attack SSH X forwarding which makes the communication secure by tunneling the x protocol.

A.3. Data compression is used to reduce the size of the file or a directory. Compressing directories help to compress the entire directory into one file so that it can be transmitted across the network via email, taken via pen drives. data compression is done in 4 ways

1.Zip

2.tar

3.tar.gz

4.TAR.BZ2

1. Compressing using zip in Linux. Zip is the most commonly used archive files which is used in both Linux, Windows and Mac OS. Compressing using the zip compresses till 60%.

Compressing using zip in Linux

zip -r archive_name.zip directory to compress

Uncompressing using zip in Linux.

unzip archive_name.zip

2. Compressing using tar in Linux. This kind of compression takes less cpu time but the compression is not much

Compressing using tar in Linux

tar -cvf archive_name.tar directory_to_compress

Uncompressing using tar in Linux

tar -xvf archive_name.tar.gz

3. compressing using tar.gz :- This kind of compression takes lot of cpu time but the compression is the highest

To compress the folder or file the syntax is

tar -zcvf archive_name.tar.gz directory_to_compress

To uncompress the folder or file the syntax is

tar -zxvf archive_name.tar.gz

To extract to another folder the command is

tar -zxvf archive_name.tar.gz -C /tmp/extract_here/

4. Compressing using the tar.bz2 :- This kind of compression is the best wasy to compress the file /folder

This takes maximum CPU time.

to compress the code is

```
# tar -jcvf archive_name.tar.bz2 directory_to_compress
```

To extract the file in the same directory the code is

```
# tar -jxvf archive_name.tar.bz2
```

To extract the file in a different directory the code is

```
# tar -jxvf archive_name.tar.bz2 -C /tmp/extract_here/
```

Answers to check your progress 2

A.1 The types of users are super user, system user, network user ,normal user, pseudo user

A.2 The command to create a super user is useradd – u 0 username

A.3 The command to create a system user is Useradd – u uid username

uid is user id which ranges from (1 – 499)

A.4 The command to create a normal user is useradd username

A.5 The command to create a pseudo user is usermod -aG sudo username

A.6 The command to create a group is

#groupadd groupname

Example #groupadd admin

A.7 The command to add a single user under a group is

#usermod –G group name username

A.8 The command to add multiple users under a group is

#gpasswd –M user1,user2,user3 groupname

A.9 The command to remove a user from a group is

#gpasswd –a username group

A.10 The command to delete a user is

#userdel username

```
[root@localhost Desktop]# userdel ram
```

#userdel-r username

A.11 The command to rename a user is

usermod –l newusername oldusername

A.12 The command to lock a user is

usermod –l username

A.13 The command to unlock a user is

usermod –u username

Answers to check your progress 3

A.1Su stands for switch user

A.2 Sudo stands for super user do

A.3 To print a file the command is

cat filename > /dev/lp

A.4 RPM stands for Red Hat Package Manager and YUM stands for Yellow dog updater, modifier

Answers to check your progress 4

A.1 Yast is a graphical user interface to which can do the following jobs :

- Install and remove software
- Set up your printer
- Configure the firewall
- Enable and disable system services
- Configure network sharing (samba)
- Format and partition your drives
- Enable NTP daemon

A.2. Webmin is a software which is used for

- Managing the server remotely from a web interface
- Managing users and groups.
- Managing soft wares and updating them.
- Monitor servers.
- Schedule backups.
- Manage services.
- Manage networking systems.
- Set cron jobs and many more.