Introduction

The purpose of this document is to provide the reader with a fast and simple introduction to using the Linux command shell and some of its basic utilities. It is assumed that the reader has zero or very limited exposure to the Linux command prompt. This document is designed to accompany an instructor-led tutorial on this subject, and therefore some details have been left out. Explanations, practical examples, and references to DOS commands are made, where appropriate.

What is Linux?

Linux (often pronounced LIH-nuhks with a short "i") is a Unix-like operating system that was designed to provide personal computer users a free or very low-cost operating system comparable to traditional and usually more expensive UNIX systems. Linux has a reputation as a very efficient and fast-performing system. Linux's kernel (the central part of the operating system) was developed by Linus Torvalds at the University of Helsinki in Finland. To complete the operating system, Torvalds and other team members made use of system components developed by members of the Free Software Foundation for the GNU Project.
Linux is the best-known and most-used open source operating system. As an operating system, Linux is software that sits underneath all of the other software on a computer, receiving requests from those programs and relaying these requests to the computer’s hardware.

The birth of Linux

On August 25, 1991, a Finn computer science student named Linus Torvalds made the following announcement to the Usenet group comp.os.minix:

> “Hello everybody out there using minix -
>
> I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready.”

Torvalds built the core of the Linux operating system, known as the kernel. A kernel alone does not make an operating system, but Stallman's GNU tools were from a project to create an operating system as well--a project that was missing a kernel to make Stallman's operating system complete. Torvalds' matching of GNU tools with the Linux kernel marked the beginning of the Linux operating system as it is known today.

Where is Linux?

One of the most noted properties of Linux is where it can be used. Linux, which began its existence as a server OS and has become useful as a desktop OS, can also be used on all of these devices. “From wristwatches to supercomputers,” is the popular description of Linux' capabilities.

The Applications and distributions *of Linux*

Linux application developers have a larger set of choices to develop their application. This allows more flexibility to build an application, but it does mean a developer will need to decide which Linux components to use. Torvalds still manages the development of the Linux kernel, but commercial and private developers contribute other software to make the whole Linux operating system. Distributions are maintained by private individuals and commercial entities. There are, at last count, over 350 distinct distributions of Linux. Linux communities come in two basic forms: developer and user communities.

Developer communities can volunteer to maintain and support whole distributions, such as the Debian or Gentoo Projects. Novell and Red hat also support community-driven versions of their products, openSUSE and Fedora, respectively.
Other developer communities focus on different applications and environments that run on Linux, such as Firefox, OpenOffice.org, GNOME, and KDE.

History of Linux

Late 1960's - Unix is developed and released in 1970's which was created by the Ken Thompson and Dennis Ritchie. It is widely adopted in business and academic circles.
In 1983 - a programmer Richard Stallman creates the GNU Project. It is an attempt at creating a Unix type operating system but composed of entirely free software.
In 1987 - Another programmer Andrew S. Tanenbaum creates Minix, a Unix like operating system for Academic use.
In 1991 - a Finnish student Linus Torvalds creates a non-commercial version of Minix and calls it Linux. The Linu is from Linus and the x is from the 'ix' part of Minix. On 25 August, 1991, Linus posted his famous message on the MINIX Newsgroup about the development of Linux.
No announcement was ever made for Linux Version 0.01. On October 5, 1991, Linus announced the first "official" version of Linux, version 0.02.
The primary focus was kernel development; none of the issues of user support, documentation, distribution, and so on had even been addressed. Today, the situation is quite different—the real excitement in the Linux world deals with graphical user environments, easy-to-install distribution packages, and high-level applications such as graphics utilities and productivity suites.

Why use Linux?

1. It's free.
2. It's a lot more secure and a lot less prone to viruses and hackers than Microsoft Windows.
3. You can do 99% of what you can do on Windows on Linux.
4. Linux has the support of a worldwide community of developers who contribute to the source code, security fixes and system enhancements.
5. Every Linux distribution offers regular updates of its packages and sources several times per year and security fixes as needed.
6. Linux systems rarely crash, and when they do, the whole system normally does not go down.
7. Linux typically does not slow down over time.
8. Linux can breathe new life into old computers.
9. Linux comes in all sizes and flavors, which offers a wide variety from which to choose the distro which will best suit our needs.
10. All Linux software is available on the Internet, so we never lose it.

What is Open Source?

The term "open source" refers to something that can be modified and shared because its design is publicly accessible. Open source code is typically created as a collaborative effort in which programmers improve upon the code and share the changes within the community. Open Source is a certification mark owned by the Open Source Initiative (OSI).

What is Open Source Software?

Open source software is software whose source code is available for modification or enhancement by anyone.

Open Source Software vs. Other software

Some software has source code that cannot be modified by anyone but the person, team, or organization; that created it and maintains exclusive control over it. This kind of software is frequently called "proprietary software" or "closed source" software, because its source code is the property of its original authors, who are the only ones legally allowed to copy or modify it. Microsoft Word and Adobe Photoshop are examples of proprietary software.

Open source software is different. Its authors make its source code available to others who would like to view that code, copy it, learn from it, alter it, or share it. LibreOffice and the GNU Image Manipulation Program are examples of open source software.

Culture of free Software

“Free software” means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. Thus, “free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer”.

A program is free software if the program's users have the four essential freedoms:

- The freedom to run the program as your wish, for any purpose (freedom 0).
- The freedom to study how the program works and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes.

What is a command *shell?*

A shell is an environment in which we can run our commands, programs, and shell scripts. There are different flavors of shells, just as there are different flavors of operating systems. Each flavor of shell has its own set of recognized commands and functions.

- A program that interprets commands
- Allows a user to execute commands by typing them manually at a terminal, or automatically in programs called *shell scripts*.
- A shell is *not* an operating system. It is a way to interface with the operating system and run commands.

Shell Prompt

The prompt, $, which is called command prompt, is issued by the shell. While the prompt is displayed, you can type a command. The shell reads your input after you press Enter. It determines the command you want executed by looking at the first word of your input.

Shell Types

There are many “shells” in both Linux and Unix. Two kinds of these numerous shells are:

1. **The Bourne shell.** If you are using a Bourne-type shell, the default prompt is the $ character.
2. **The C shell (csh)**
    - C shell is the UNIX shell created by Bill Joy at the University of California as an alternative to UNIX's original shell, the Bourne shell .
    - If you are using a C-type shell, the default prompt is the % character. The C shell program name is ***csh***.
    - The C shell was invented for programmers who prefer a syntax similar to that of the C programming language.

- The other popular member of the C shell family is called *tcsh* (for Tab C shell) and is an extended version of C shell. Some of **tcsh's** added features are: enhanced *history substitution* , spelling correction, and *word completion*

There are again various subcategories for Bourne Shell which are listed as follows:

- Bourne shell (**sh**)
- Korn shell (**ksh**)
  - The Korn shell is the UNIX shell (command execution program, often called a *command interpreter*) that was developed by David Korn of Bell Labs as a comprehensive combined version of other major UNIX shells.
  - Incorporating all the features of C shell (*csh*) and Tab C-shell **(** ***tcsh***) with the script language features similar to that of the Bourne shell , the Korn shell is considered the most efficient shell.
  - The Korn shell is considered a member of the Bourne shell family and uses as its shell prompt (character displayed to indicate readiness for user input) the *$* symbol.
  - Because it is the easiest shell to use, inexperienced users usually prefer the Korn shell
  - Korn shell is developed many years before the emergence of the BASH shell. Because it is older than BASH, it has less resources, and it also attracts limited computer users.
- Bourne Again shell **(bash**)
- POSIX shell (**sh**)

What is BASH?

- BASH = **B**ourne **A**gain **SH**ell
- Bash is a shell written as a free replacement to the standard Bourne Shell (/bin/sh)
- Originally written by Steve Bourne for UNIX systems.
- It has all of the features of the original Bourne Shell, plus additions that make it easier to program with and use from the command line.
- Since it is Free Software, it has been adopted as the default shell on most Linux systems.
- Bash is a newer shell compared to the KSH shell. Bash also acts as an extension of the Korn shell.

How is BASH different from the DOS command prompt?

Case Sensitivity:	In Linux/UNIX, commands and filenames are case sensitive, meaning that typing "EXIT" instead of the proper "exit" is a mistake.
"\"vs."/":	In DOS, the forward-slash "/" is the command argument delimiter, While the backslash "\" is a directory separator. In Linux/UNIX, the "/" is the directory separator, and the "\" is an escape character.
Filenames:	The DOS world uses the "eight dot three" filename convention, meaning that all files followed a format that allowed up to 8 characters in the filename, followed by a period ("dot"), followed by an option extension, up to 3 characters long (e.g. FILENAME.TXT). In UNIX/Linux, there is no such thing as a file extension. Periods can be placed at any part of the filename, and "extensions" may be interpreted differently by all programs, or not at all.

Special Characters

Before we continue to learn about Linux shell commands, it is important to know that there are many symbols and characters that the shell interprets in special ways.
This means that certain typed characters: a) cannot be used in certain situations, b) may be used to perform special operations, or, c) must be "escaped" if you want to use them in a normal way.

character	*Description*
\	Escape character. If you want to reference a special character, you must "escape" it with a backslash first. Example: touch /tmp/filename*
/	Directory separator, used to separate a string of directory names. Example: /usr/src/linux
.	Current directory. Can also "hide" files when it is the first character in a filename.
..	Parent directory
~	User's home directory
*	Represents 0 or more characters in a filename, or by itself, all files in a directory. Example: pic*2002 can represent the files pic2002, picJanuary2002, picFeb292002, etc.
?	Represents a single character in a filename. Example: hello?.txt can represent hello1.txt, helloz.txt, but not hello22.txt
[ ]	Can be used to represent a range of values, e.g. [0-9], [A-Z], etc. Example: hello[0-2].txt represents the names hello0.txt, hello1.txt, and hello2.txt
|	"Pipe". Redirect the output of one command into another command. Example: ls | more
>	Redirect output of a command into a new file. If the file already exists, over-write it. Example: ls > myfiles.txt
>>	Redirect the output of a command onto the end of an existing file. Example: echo "Mary 555-1234" >> phonenumbers.txt
<	Redirect a file as input to a program. Example: more < phonenumbers.txt
;	Command separator. Allows you to execute multiple commands on a single line. Example: cd /var/log ; less messages
&&	Command separator as above, but only runs the second command if the first one Finished without errors. Example: cd /var/logs && less messages
&	Execute a command in the background, and immediately get your shell back. Example: find / -name core > /tmp/corefiles.txt &

Executing Commands

The Command PATH:

- Most common commands are located in your shell's "PATH", meaning that you can just type the name of the program to execute it. Example: Typing " ls" will execute the " ls" command.
- Your shell's "PATH" variable includes the most common program locations, such as

/bin, /usr/bin, /usr/X11R6/bin, and others.

- To execute commands that are not in your current PATH, you have to give the complete location of the command.

  Examples: /home/bob/myprogram

  ./program (Execute a program in the current directory)

  ~/bin/program (Execute program from a personal bin directory)

Command Syntax:

- Commands can be run by themselves, or you can pass in additional arguments to make them do different things. Typical command syntax can look something like this:

```
Command [-argument] [-argument] [--argument] [file]
```

- Examples:

`ls`	List files in current directory
`ls –l`	Lists files in "long" format
`ls -l –color`	As above, with colorized output
`cat filename`	Show contents of a file

Getting Help

When you're stuck and need help with a Linux command, help is usually only a few keystrokes away! Help on most Linux commands is typically built right into the commands themselves, available through online help programs ("man pages" and "info pages"), and of course online.

Using a Command's Built-In Help

Many commands have simple "help" screens that can be invoked with special command flags. These flags usually look like "`–h`" or "`--help`".
Example: grep --help

Online Manuals: "Man Pages"

The best source of information for most commands can be found in the online manual pages, known as "man pages" for short. To read a command's man page, type "man command".

Examples:		
	`man ls`	Get help on the "ls" command.
	`man man`	A manual about how to use the manual!

To search for a particular word within a man page, type "/word". To quit from a man page, just type the "Q" key.

Sometimes, you might not remember the name of Linux command and you need to search for it. For example, if you want to know how to change a file's permissions, you can search the man page descriptions for the word "permission" like this: `man –k permission`
If you look at the output of this command, you will find a line that looks something like:

```
Chmod           (1)- change file access permissions
```

Navigating the Linux Filesystem

The Linux filesystem is a tree-like hierarchy, hierarchy of directories and files. At the base of the filesystem is the "/" directory, otherwise known as the "root" (not to be confused with the root user). Unlike DOS or Windows filesystems that have multiple "roots", one for each disk drive, the

Linux filesystem mounts all disks somewhere underneath the / filesystem. The following table describes many of the most common Linux directories.

The Linux Directory Layout

Directory	Description
	The nameless base of the filesystem. All other directories, files, drives, and devices are attached to this root. Commonly (but incorrectly) referred to as the “slash” or “/” directory. The “/” is just a directory separator, not a directory itself.
/bin	Essential command binaries (programs) are stored here (bash, ls, mount, tar, etc.)
/boot	Static files of the boot loader.
/dev	Device files. In Linux, hardware devices are acceessd just like other files, and they are kept under this directory.
/etc	Host-specific system configuration files.
/home	Location of users' personal home directories (e.g. /home/susan).
/lib	Essential shared libraries and kernel modules.
/proc	Process information pseudo-filesystem. An interface to kernel data structures
/root	The root (superuser) home directory.
/sbin	Essential system binaries (fdisk, fsck, init, etc).
/tmp	Temporary files. All users have permission to place temporary files here.
/usr	The base directory for most shareable, read-only data (programs, libraries, documentation, and much more).
/usr/bin	Most user programs are kept here (cc, find, du, etc.).
/usr/include	Header files for compiling C programs.
/usr/lib	Libraries for most binary programs.
/usr/local	“Locally” installed files. This directory only really matters in environments where files are stored on the network. Locally-installed files go in /usr/local/bin, /usr/local/lib, etc.). Also often used for software packages installed from source, or software not officially shipped with the distribution.
/usr/sbin	Non-vital system binaries (lpd, useradd, etc.)
/usr/share	Architecture-independent data (icons, backgrounds, documentation, terminfo, man pages, etc.).
/usr/src	Program source code. E.g. The Linux Kernel, source RPMs, etc.
/usr/X11R6	The X Window System.
/var	Variable data: mail and printer spools, log files, lock files, etc.

Commands for Navigating the Linux Filesystems

The first thing you usually want to do when learning about the Linux filesystem is take some time to look around and see what's there! These next few commands will: a) Tell you where you are, b) take you somewhere else, and c) show you what's there. The following table describes the basic operation of the pwd, cd, and ls commands, and compares them to certain DOS commands that you might already be familiar with.

Linux command	DOS command	Description
pwd	Cd	“Print Working Directory”. Shows the current location in the directory tree.

cd	cd, chdir	"Change Directory". When typed all by itself, it returns you to your home directory.
cd directory	cd directory	Change into the specified directory name. Example: cd /usr/src/linux
cd ~		"~" is an alias for your home directory. It can be used as a shortcut to your "home", or other directories relative to your home.
cd ..	cd..	Move up one directory. For example, if you are in /home/vic and you type "cd ..", you will end up in /home.
cd -		Return to previous directory. An easy way to get back to your previous location!
ls	dir /w	List all files in the current directory, in column format.
ls directory	dir directory	List the files in the specified directory. Example: ls /var/log
ls -l	Dir	List files in "long" format, one file per line. This also shows you additional info about the file, such as ownership, permissions, date, and size.
ls -a	dir /a	List all files, including "hidden" files. Hidden files are those files that begin with a ".", e.g. The .bash_history file in your home directory.
ls -ld directory		A "long" list of "directory", but instead of showing the directory contents, show the directory's detailed information. For example, compare the output of the following two commands: ls -l /usr/bin ls -ld /usr/bin
ls /usr/bin/d*	dir d*.*	List all files whose names begin with the letter "d" in the /usr/bin directory.

Piping and Re-Direction

Before we move on to learning even more commands, let's side-track to the topics of piping and re-direction. The basic UNIX philosophy, therefore by extension the Linux philosophy, is to have many small programs and utilities that do a particular job very well. It is the responsibility of the programmer or user to combine these utilities to make more useful command sequences.

Piping Commands Together

The pipe character, "|", is used to chain two or more commands together. The output of the first command is "piped" into the next program, and if there is a second pipe, the output is sent to the third program, etc. For example: `ls -la /usr/bin | less`
In this example, we run the command "ls -la /usr/bin", which gives us a long listing of all of the files in /usr/bin. Because the output of this command is typically very long, we pipe the output to a program called "less", which displays the output for us one screen at a time.

Redirecting Program Output to Files

There are times when it is useful to save the output of a command to a file, instead of displaying it to the screen. For example, if we want to create a file that lists all of the MP3 files in a directory, we can do something like this, using the ">" redirection character:

```
     ls -l /home/vic/MP3/*.mp3 > mp3files.txt
```

A similar command can be written so that instead of creating a new file called `mp3files.txt`, we can append to the end of the original file:

```
     ls -l /home/vic/extraMP3s/*.mp3 >> mp3files.txt
```

Other Linux Commands

The following sections describe many other commands that you will find on most Linux systems. I can't possibly cover the details of all of these commands in this document, so don't forget that you can check the "man pages" for additional information. Not all of the listed commands will be available on all Linux or UNIX distributions.

Working With Files and Directories

These commands can be used to: find out information about files, display files, and manipulate them in other ways (copy, move, delete).

Linux Command	**DOS Command**	**Description**
file		Find out what kind of file it is. For example, "file /bin/ls" tells us that it is a Linux executable file.
cat	type	Display the contents of a text file on the screen. For example: cat mp3files.txt would display the file we created in the previous section.
head		Display the first few lines of a text file. Example: head /etc/services
tail		Display the last few lines of a text file. Example: tail /etc/services
tail -f		Display the last few lines of a text file, and then output appended data as the file grows (very useful for following log files!). Example: tail -f /var/log/messages
cp	copy	Copies a file from one location to another. Example: cp mp3files.txt /tmp (copies the mp3files.txt file to the /tmp directory)
mv	rename, ren, move	Moves a file to a new location, or renames it. For example: mv mp3files.txt /tmp (copy the file to /tmp, and delete it from the original location)
rm	del	Delete a file. Example: rm /tmp/mp3files.txt
mkdir	md	Make Directory. Example: mkdir /tmp/myfiles/
rmdir	rd, rmdir	Remove Directory. Example: rmdir /tmp/myfiles/

Finding Things

The following commands are used to find files. "ls" is good for finding files if you already know approximately where they are, but sometimes you need more powerful tools such as these:

Linux Command	Description
which	Shows the full path of shell commands found in your path. For example, if you want to know exactly where the “grep” command is located on the filesystem, you can type “which grep”. The output should be something like: /bin/grep
whereis	Locates the program, source code, and manual page for a command (if all information is available). For example, to find out where “ls” and its man page are, type: “whereis ls” The output will look something like: ls: /bin/ls/usr/share/man/man1/ls.1.gz
locate	A quick way to search for files anywhere on the filesystem. For example, you can find all files and directories that contain the name “mozilla” by typing: locate mozilla
find	A very powerful command, but sometimes tricky to use. It can be used to search for files matching certain patterns, as well as many other types of searches. A simple example is: find . -name *mp3 This example starts searching in the current directory “.” and all sub- directories, looking for files with “mp3” at the end of their names.

Informational Commands

The following commands are used to find out some information about the user or the system.

Linux Command	*Explanation*
ps	Lists currently running process (programs).
w	Show who is logged on and what they are doing.
id	Print your user-id and group id's
df	Report filesystem disk space usage (“Disk Free” is how I remember it)
du	Disk Usage in a particular directory. “du -s” provides a summary for the current directory.
top	Displays CPU processes in a full-screen GUI. A great way to see the activity on your computer in real-time. Type “Q” to quit.
free	Displays amount of free and used memory in the system.
cat /proc/cpuinfo	Displays information about your CPU.
cat /proc/meminfo	Display lots of information about current memory usage.
uname -a	Prints system information to the screen (kernel version, machine type, etc.)

Other Utilities

Here are some other commands that are useful to know.

Linux Command	*Description*
clear	Clear the screen
echo	Display text on the screen. Mostly useful when writing shell scripts. For example: echo “Hello World”
more	Display a file, or program output one page at a time. Examples: more mp3files.txt ls -la | more
less	An improved replacement for the “more” command. Allows you to scroll Backwards as well as forwards.
grep	Search for a pattern in a file or program output. For example, to find out which TCP network port is used by the “nfs” service, you can do this: grep “nfs” /etc/services This looks for any line that contains the string “nfs” in the file “/etc/services” and displays only those lines.
lpr	Print a file or program output. Examples: lpr mp3files.txt - Print the mp3files.txt file ls -la | lpr - Print the output of the “ls -la” command.
sort	Sort a file or program output. Example: sort mp3files.txt
su	“Switch User”. Allows you to switch to another user's account temporarily. The default account to switch to is the root/superuser account. Examples: su - Switch the root account su - - Switch to root, and log in with root's environment su larry - Switch to Larry's account

Shortcuts to Make it all Easier!

Shortcut	*Description*
Up/Down Arrow Keys	Scroll through your most recent commands. You can scroll back to an old command, hit Enter, and execute the command without having to re-type it.
“history” command	Show your complete command history.
TAB Completion	If you type a partial command or filename that the shell recognizes, you can have it automatically completed for you if you press the TAB key. Try typing the first few characters of your favorite Linux command, then hit TAB a couple of times to see
Complete recent commands with “!”	Try this: Type “!” followed by the first couple of letters of a recent command and press ENTER! For example, type: find /usr/bin -type f -name m* ...and now type: !fi
Search your command history with CTRL-R	Press CTRL-R and then type any portion of a recent command. It will search the commands for you, and once you find the command you want, just press ENTER.
Scrolling the screen with Shift-PageUp and Page Down	Scroll back and forward through your terminal.

1. Some Useful Linux Commands

- **/** - root directory
- **./** - current directory
- **./command_name** - run a command in the current directory when the current directory is not on the path
- **../** - parent directory
- **~** - home directory
- **$** - typical prompt when logged in as ordinary user
- **#** - typical prompt when logged in as root or superuser
- **!** - repeat specified command
- **!!** - repeat previous command
- **^^** - repeat previous command with substitution
- **&** - run a program in background mode
- **[Tab][Tab]** - prints a list of all available commands. This is just an example of autocomplete with no restriction on the first letter.
- **x[Tab][Tab]** - prints a list of all available completions for a command, where the beginning is ``x''
- **[Alt][Ctrl][F1]** - switch to the first virtual text console
- **[Alt][Ctrl][Fn]** - switch to the nth virtual text console. Typically, there are six on a Linux PC system.
- **[Alt][Ctrl][F7]** - switch to the first GUI console, if there is one running. If the graphical console freezes, one can switch to a nongraphical console, kill the process that is giving problems, and switch back to the graphical console using this shortcut.
- **[ArrowUp]** - scroll through the command history (in bash)
- **[Shift][PageUp]** - scroll terminal output up. This also works at the login prompt, so you can scroll through your boot messages.
- **[Shift][PageDown]** - scroll terminal output down
- **[Ctrl][Alt][+]** - switch to next X server resolution (if the server is set up for more than one resolution)
- **[Ctrl][Alt][-]** - change to previous X server resolution
- **[Ctrl][Alt][BkSpc]** - kill the current X server. Used when normal exit is not possible.
- **[Ctrl][Alt][Del]** - shut down the system and reboot
- **[Ctrl] c** - kill the current process
- **[Ctrl] d** - logout from the current terminal
- **[Ctrl] s** - stop transfer to current terminal
- **[Ctrl] q** - resume transfer to current terminal. This should be tried if the terminal stops responding.
- **[Ctrl] z** - send current process to the background
- **[Leftmousebutton]** - Hold down left mouse button and drag to highlight text. Releasing the button copies the region to the text buffer under X and (if gpm is installed) in console mode.
- **[Middlemousebutton]** - Copies text from the text buffer and inserts it at the cursor location. With a two-button mouse, click on both buttons simultaneously. It is necessary for three-button emulation to be enabled, either under gpm or in XF86Config.

2. Typical Dot Files

- **.bash_logout** - file executed by bash shell on logout
- **.bash_profile** - initialization of bash shell run only on login. Bash looks first for a .bash_profile file when started as a login shell or with the -login option. If it does not find .bash_profile, it looks for .bash_login. If it doesn't find that, it looks for .profile. System-wide functions and aliases go in /etc/bashrc and default environment variables go in /etc/profile.
- **.login** - initialization file when user logs in

3. Useful Files

- **/boot/vmlinuz** - the typical location and name of the Linux kernel. In the Slackware distribution, the kernel is located at /vmlinuz.
- **/dev/fd0** - first floppy disk drive
- **/dev/fd0H1440** - driver for the first floppy drive in high density mode. Generally, this is invoked when formatting a floppy drive for a particular density. Slackware comes with drivers that allow for formatting a 3.5" diskette with up to 1.7MB of space. Red Hat and Mandrake do not contain these device driver files by default.
- **/dev/fd1** - second floppy disk drive
- **/dev/hda** - first IDE hard drive
- **/dev/hdc** - on many machines, the IDE cdrom drive. Most often, there is a symbolic link called /dev/cdrom which is just a link to the true cdrom driver file.
- **/dev/null** - used when you want to send output into oblivion
- **/etc/aliases** - file containing aliases used by sendmail and other MTAs (mail transport agents). After updating this file, it is necessary to run the newaliases utility for the changes to be passed to sendmail.
- **/etc/bashrc** - system-wide default functions and aliases for the bash shell
- **/etc/conf.modules** - aliases and options for configurable modules
- **/etc/crontab** - shell script to run different commands periodically (hourly, daily, weekly, monthly, etc.)
- **/etc/DIR_COLORS** - used to store colors for different file types when using ls command. The dircolors command uses this file when there is not a .dir_colors file in the user's home directory. Used in conjunction with the eval command (see below).
- **/etc/exports** - specifies hosts to which file systems can be exported using NFS. Man exports contain information on how to set up this file for remote users.
- **/etc/fstab** - contains information on partitions and filesystems used by system to mount different partitions and devices on the directory tree
- **/etc/HOSTNAME** - stores the name of the host computer
- **/etc/hosts** - contains a list of host names and absolute IP addresses.
- **/etc/hosts.allow** - hosts allowed (by the tcpd daemon) to access Internet services
- **/etc/hosts.deny** - hosts forbidden (by the tcpd daemon) to access Internet services
- **/etc/group** - similar to /etc/passwd but for groups
- **/etc/inetd.conf** - configures the inetd daemon to tell it what TCP/IP services to provide (which daemons to load at boot time). A good start to securing a Linux box is to turn off these services unless they are necessary.
- **/etc/inittab** - runs different programs and processes on startup. This is typically the program which is responsible for, among other things, setting the default runlevel, running the rc.sysinit script contained in /etc/rc.d, setting up virtual login terminals, bringing down the system in an orderly fashion in response to [Ctrl][Alt][Del], running the rc script in /etc/rc.d, and running xdm for a graphical login prompt (only if the default runlevel is set for a graphical login).
- **/etc/issue** - pre-login message. This is often overwritten by the /etc/rc.d/rc.S script (in Slackware) or by the /etc/rc.d/rc.local script (in Mandrake and Red Hat, and perhaps other rpm-based distributions). The relevant lines should be commented out (or changed) in these scripts if a custom pre-login message is desired.
- **/etc/lilo.conf** - configuration file for lilo boot loader
- **/etc/motd** - message of the day file, printed immediately after login. This is often overwritten by /etc/rc.d/rc.S (Slackware) or /etc/rc.d/rc.local (Mandrake/Red Hat) on startup. See the remarks in connection with /etc/issue.
- **/etc/mtab** - shows currently mounted devices and partitions and their status

- **/etc/passwd** - contains passwords and other information concerning users who are registered to use the system. For obvious security reasons, this is readable only by root. It can be modified by root directly, but it is preferable to use a configuration utility such as passwd to make the changes. A corrupt /etc/passwd file can easily render a Linux box unusable.
- **/etc/printcap** - shows the setup of printers
- **/etc/profile** - sets system-wide defaults for bash shell. It is this file in Slackware that sets up the DIR_COLORS environment variable for the color ls command. Also sets up other system-wide environment variables.
- **/etc/resolv.conf** - contains a list of domain name servers used by the local machine
- **/etc/securetty** - contains a list of terminals on which root can login. For security reasons, this should not include dialup terminals.
- **/etc/termcap** - ASCII database defining the capabilities and characteristics of different consoles, terminals, and printers
- /etc/**X11/XF86Config** - X configuration file. The location in Slackware is /etc/XF86Config.
- **/proc/cpuinfo** - cpu information
- **/proc/filesystems** - prints filesystems currently in use
- **/proc/interrupts** - prints interrupts currently in use
- **/proc/ioports** - contains a list of the i/o addresses used by various devices connected to the computer
- **/proc/kcore** - The command ls -l /proc/kcore will give the amount of RAM on the computer. It's also possible to use the free command to get the same information (and more).
- **/proc/version** - prints Linux version and other info
- **/var/log/messages** - used by syslog daemon to store kernel boot-time messages
- **/var/log/lastlog** - used by system to store information about last boot
- **/var/log/wtmp** - contains binary data indicating login times and duration for each user on system

4. Important Directories

Different distributions have different directory structures, despite attempts at standardization such as the the Linux Filesystem Hierarchy Standard (FHS) organization.

- **/bin** - essential UNIX commands such as ls, etc. Should contain all binaries needed to boot the system or run it in single-user mode
- **/boot** - files used during booting and possibly the kernel itself are stored here
- **/dev** - contains device files for various devices on system
- **/etc** - files used by subsystems such as networking, NFS, and mail. Includes tables of disks to mount, processes to run on startup, etc.
- **/etc/profile.d** - contains scripts that are run by /etc/profile upon login.
  - **/etc/rc.d** - contains a number of shell scripts that are run on bootup at different run levels. There is also typically an rc.inet1 script to set up networking (in Slackwar), an rc.modules script to load modular device drivers, and an rc.local script that can be edited to run commands desired by the administrator, along the lines of autoexec.bat in DOS.
- **/etc/rc.d/init.d** - contains most of the initialization scripts themselves on an rpm-based system.
- **/etc/rc.d/rc*.d** - where ``*'' is a number corresponding to the default run level. Contains files for services to be started and stopped at that run level. On rpm-based systems, these files are symbolic links to the initialization scripts themselves, which are in /etc/rc.d/init.d.
- **/etc/skel** - directory containing several example or skeleton initialization shells. Often contains subdirectories and files used to populate a new user's home directory.
- /etc/**X11** - configuration files for the X Window system
- **/home** - home directories of individual users

- **/lib** - standard shared library files
- **/lib/modules** - modular device driver files, most with .o extensions
- **/mnt** - typical mount point for many user-mountable devices such as floppy drives, cd-rom readers, etc. Each device is mounted on a subdirectory of /mnt.
- **/proc** - virtual file system that provides a number of system statistics
- **/root** - home directory for root
- **/sbin** - location of binaries used for system administration, configuration, and monitoring
- **/tmp** - directory specifically designed for programs and users to store temporary files.
- **/usr** - directory containing a number of subdirectory with programs, libraries, documentation, etc.
- **/usr/bin** - contains most user commands. Should not contain binaries necessary for booting the system, which go in /bin. The /bin directory is generally located on the same disk partition as /, which is mounted in read-only mode during the boot process. Other filesystems are only mounted at a later stage during startup, so putting binaries essential for boot here is not a good idea.
- **/usr/info** - primary location of the GNU info system files
- **/usr/lib** - standard library files such as libc.a. Searched by the linker when programs are compiled.
- **/usr/lib/X11** - X Window system distribution
- **/usr/local/bin** - yet another place to look for comon executables
- **/usr/man** - location of manual page files
- **/usr/sbin** - other commands used by superuser for system administration
- **/usr/share** - contains subdirectories where many installed programs have configuration, setup and auxiliary files
- **/var** - administrative files such as log files, used by various utilities
- **/var/log/packages** - contains files, each of which has detailed information on an installed package in Slackware. The same file can also be found at /var/adm/packages, since the adm subdirectory is a symbolic link to log. Each package file contains a short description plus a list of all installed files.
- **/var/log/scripts** - package installation scripts in Slackware are stored here. You can inspect these scripts to see what special features are included in individual packages.
- **/var/spool** - temporary storage for files being printed, mail that has not yet been picked up, etc.

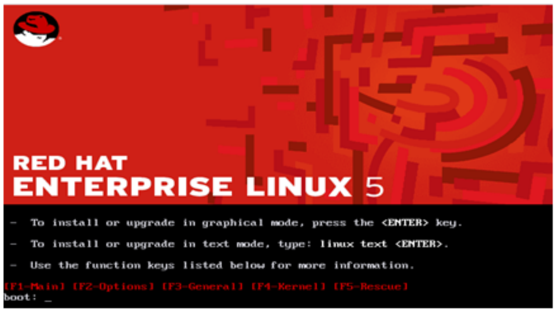
“The Linux philosophy is "laugh in the face of danger". Oops. Wrong one. "Do it yourself". That's it.”

How to Install Red Hat Linux – Complete Step by Step Guide

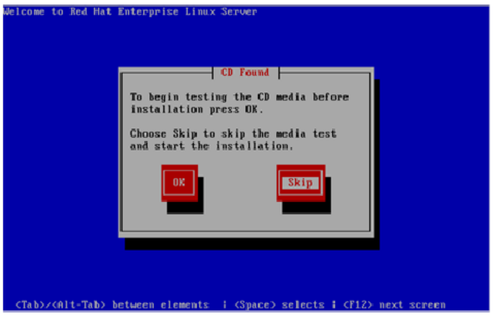
As the open source revolution grows around the world, more and more people are starting to switch over to the Linux Operating System and pre-eminent of all the Linux OS is the Red Hat Linux, owned and distributed by the Red Hat Inc. Each Linux installation has its own setup utility, every one vastly different from all the others. Here are the steps to easily install red hat Linux:

Step 1– Insert the Red Hat Linux DVD into the DVD-drive of your computer. As soon as the following screen_1 pops up, press ‘Enter’ to install Red Hat Enterprise Linux (RHEL) through GUI mode.

Step 2– RHEL installer would then prompt you conduct a check as to whether the CD media from which you’re installing is functioning correctly or not. Choose ‘Skip’, press enter and the installation would begin.



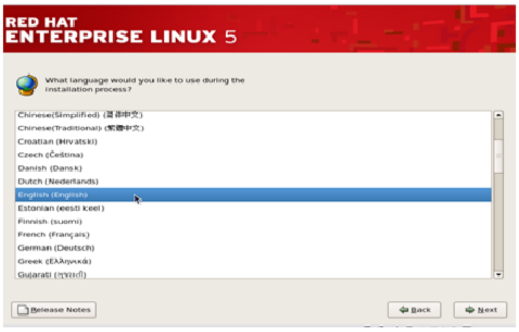
```
-  To install or upgrade in graphical mode, press the <ENTER> key.
-  To install or upgrade in text mode, type: linux text <ENTER>.
-  Use the function keys listed below for more information.
[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]
boot: _
```



```
Welcome to Red Hat Enterprise Linux Server

CD Found

To begin testing the CD media before
installation press OK.

Choose Skip to skip the media test
and start the installation.

OK        Skip

<Tab>/<Alt-Tab> between elements  | <Space> selects | <F12> next screen
```

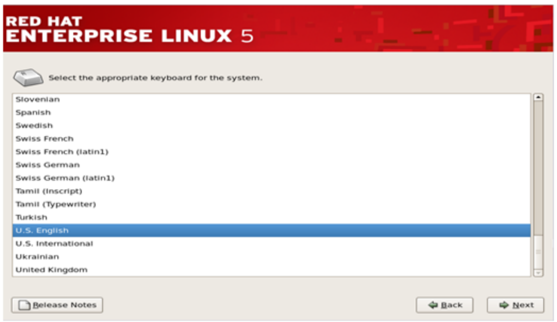
Step 3– Next, we need to select the language- English or any other language as per our preference, and then press ‘Next’.

Step 4– In this step, the RHEL installer would ask you about the appropriate type of keyboard for the system. We take the ‘US English’ keyboard; you can pick any other option depending on the type of your keyboard. Then press ‘Next’ to move to the next step.



What language would you like to use during the installation process?

- Chinese(Simplified) (简体中文)
- Chinese(Traditional) (繁體中文)
- Croatian (Hrvatski)
- Czech (Čeština)
- Danish (Dansk)
- Dutch (Nederlands)
- English (English)
- Estonian (eesti keel)
- Finnish (suomi)
- French (Français)
- German (Deutsch)
- Greek (Ελληνικά)
- Gujarati (ગુજરાતી)

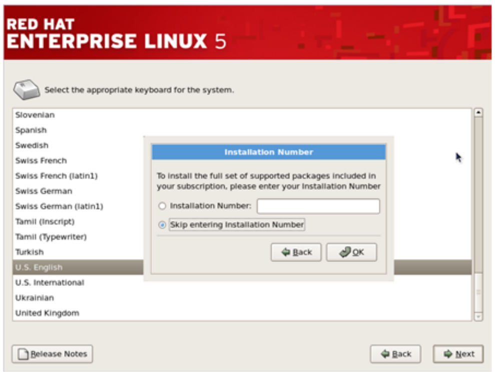


Select the appropriate keyboard for the system.

- Slovenian
- Spanish
- Swedish
- Swiss French
- Swiss French (latin1)
- Swiss German
- Swiss German (latin1)
- Tamil (Inscript)
- Tamil (Typewriter)
- Turkish
- U.S. English
- U.S. International
- Ukrainian
- United Kingdom

Step 5– Next, the installer would ask for an ‘installation number’ if you wish to install full set of Red Hat functionalities. Enter the installation number and press ‘OK’ if you have an officially licensed installation number (for corporate clients that buy Red Hat’s backup support and full features). Others can select ‘Skip entering installation number’ and press ‘OK’ to proceed. RHEL would show a warning message, press ‘Skip’ in it to continue.

Step 6– The Red Hat installer would then require you to create partitions in your computer’s hard disk for the installation. You can do it in four ways but the simplest way is to select ‘Use free space on selected drives and create default layout’ as this option will not affect any other OS residing in your system. Check the ‘review and modify portioning layout’ to create partitions and click next.



Select the appropriate keyboard for the system.

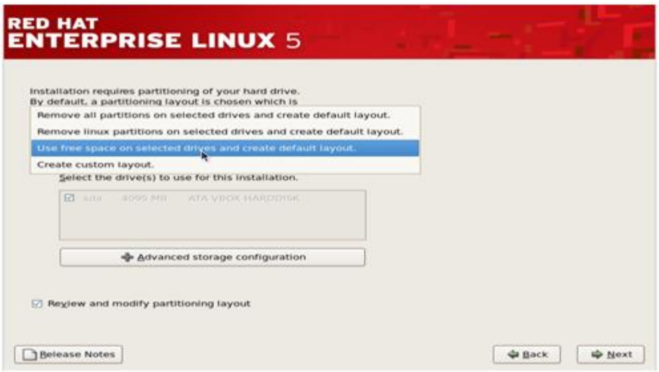
Installation Number

To install the full set of supported packages included in your subscription, please enter your Installation Number

○ Installation Number:

● Skip entering Installation Number

Back   OK



Installation requires partitioning of your hard drive.
By default, a partitioning layout is chosen which is

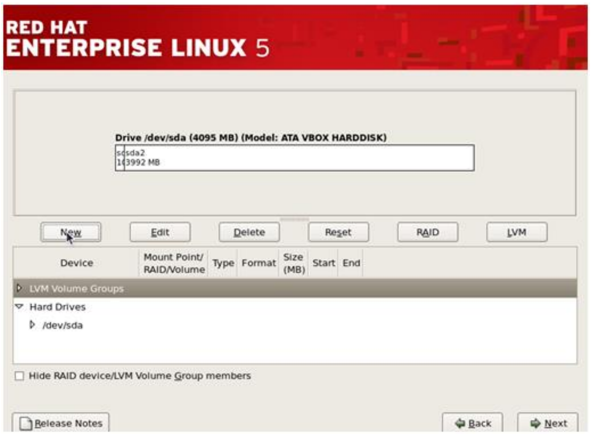
- Remove all partitions on selected drives and create default layout.
- Remove linux partitions on selected drives and create default layout.
- Use free space on selected drives and create default layout.
- Create custom layout.

Select the drive(s) to use for this installation.

☑ sda 4095 MB ATA VBOX HARDDISK

☑ Review and modify partitioning layout

Step 7– In this step you must create the required system partitions and mount points such as ‘/’, ‘/home’, ‘swap’ etc. Which are required for the Linux’s proper functioning.

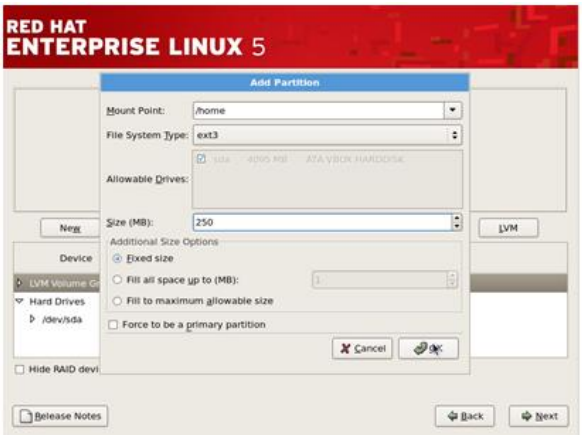


Drive /dev/sda (4095 MB) (Model: ATA VBOX HARDDISK)

sdsda2
1(3992 MB

Device	Mount Point/ RAID/Volume	Type	Format	Size (MB)	Start	End
▷ LVM Volume Groups						
▽ Hard Drives						
▷ /dev/sda						

□ Hide RAID device/LVM Volume Group members



Add Partition

Mount Point: /home

File System Type: ext3

Allowable Drives: ☑ sda 4095 MB ATA VBOX HARDDISK

Size (MB): 250

Additional Size Options

● Fixed size

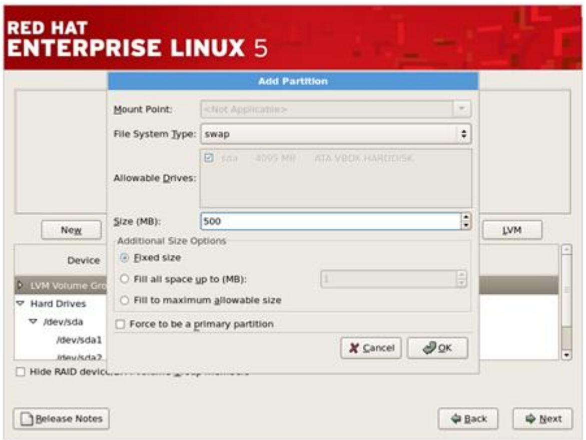
○ Fill all space up to (MB): 1

○ Fill to maximum allowable size

□ Force to be a primary partition

Then, select /home in the mount point and choose ‘ext3’ as the file system and give the desired size for it and then click ‘OK’. Similarly also create /boot and /var.

Also, create a swap partition by clicking on ‘New’ and then choosing the filesystem as ‘swap’ and also give the size of Swap partition.(Usually size of swap partition SHOULD BE twice the size of RAM available to the system but you can keep its size less than that too)



Add Partition

Mount Point: <Not Applicable>

File System Type: swap

Allowable Drives: ☑ sda 4095 MB ATA VBOX HARDDISK

Size (MB): 500

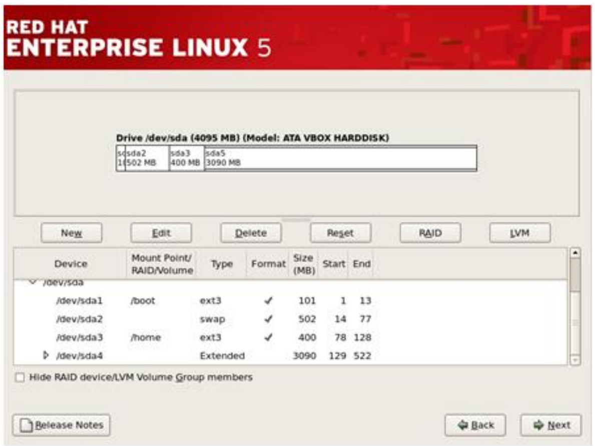
Additional Size Options

● Fixed size

○ Fill all space up to (MB): 1

○ Fill to maximum allowable size

□ Force to be a primary partition



Drive /dev/sda (4095 MB) (Model: ATA VBOX HARDDISK)

sdsda2 1(502 MB | sda3 400 MB | sda5 3090 MB

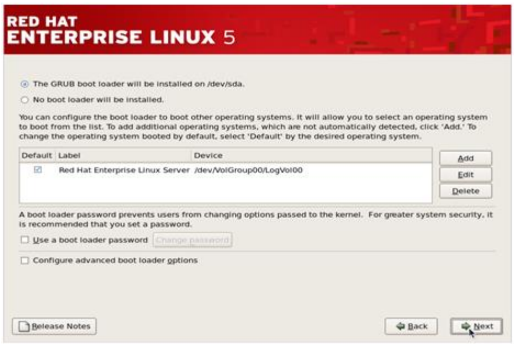
Device	Mount Point/ RAID/Volume	Type	Format	Size (MB)	Start	End
▽ /dev/sda						
/dev/sda1	/boot	ext3	✓	101	1	13
/dev/sda2		swap	✓	502	14	77
/dev/sda3	/home	ext3	✓	400	78	128
▷ /dev/sda4		Extended		3090	129	522

□ Hide RAID device/LVM Volume Group members

Once you have made all the desired partitions and given their mount points, click ‘Next’ to continue installation.

Step 8– This step pertains to the default OS that will be loaded by the GRUB loader. (Note- If you have multiple Operating Systems installed, you would see multiple options here and you have to check in front of the OS name that you want to be loaded by default when the system is started.)
Click ‘Next’ to continue.

Step 9– This step pertains to the network settings of the Linux system that you are going to install. You can select the Ethernet devices through which the system would communicate with other devices in the network.
You can also provide the hostname, Gateway address and DNS address to the system during this step. (However it’s better to adjust these settings once the system has been fully installed).



● The GRUB boot loader will be installed on /dev/sda.
○ No boot loader will be installed.

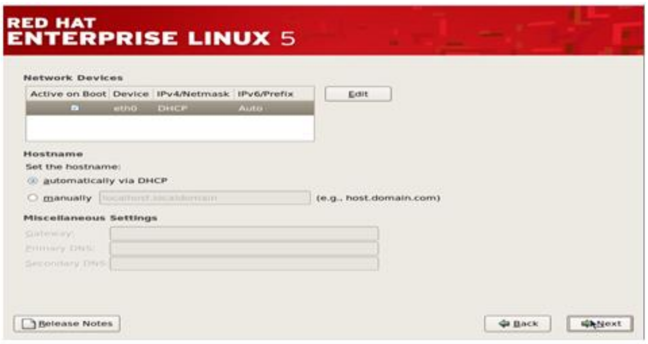
You can configure the boot loader to boot other operating systems. It will allow you to select an operating system to boot from the list. To add additional operating systems, which are not automatically detected, click 'Add.' To change the operating system booted by default, select 'Default' by the desired operating system.

Default	Label	Device
☑	Red Hat Enterprise Linux Server	/dev/VolGroup00/LogVol00

A boot loader password prevents users from changing options passed to the kernel. For greater system security, it is recommended that you set a password.

□ Use a boot loader password

□ Configure advanced boot loader options



Network Devices

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix
☑	eth0	DHCP	Auto

Hostname

Set the hostname:

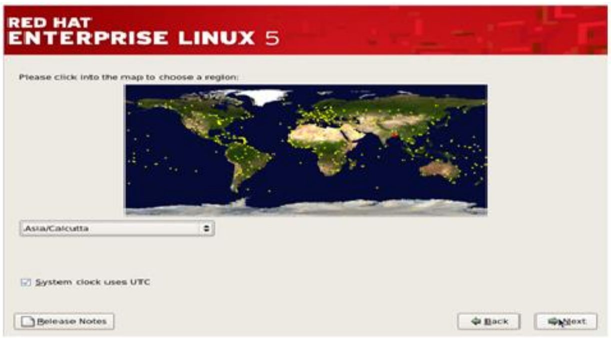
● automatically via DHCP
○ manually localhost.localdomain (e.g., host.domain.com)

Miscellaneous Settings

Gateway:
Primary DNS:
Secondary DNS:

Step 10– The next step is to adjust the system clock to your particular time zone. Select your time zone and then click ‘Next’.

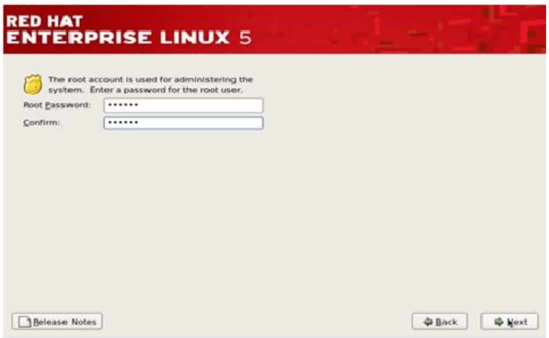
Step 11 – This is a very important step that deals with the root (super-user) password for the system. Type the password and confirm it and then click next.



Please click into the map to choose a region:

Asia/Calcutta

☑ System clock uses UTC



The root account is used for administering the system. Enter a password for the root user.

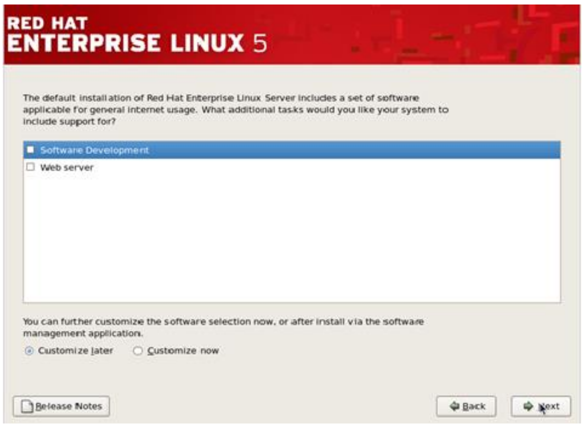
Root Password: ••••••

Confirm: ••••••

Step 12 – The RHEL installer would then prompt you about if you wish to install some extra ‘Software Development’ or ‘Web Server’ features. By default, keep it at ‘Customize later’ and press ‘Next’.

Step 13– This next step will initiate the installation of Red Hat Linux, press ‘Next’ to begin the process.

Step 14– Upon the completion of installation you should the following screen. Press Reboot and you’d be ready to use your newly installed Red Hat Linux OS.

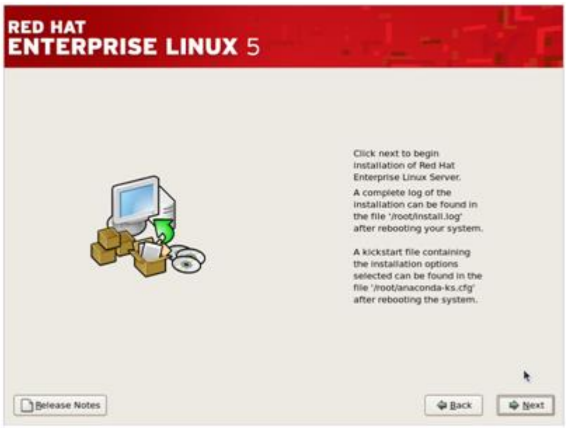


The default installation of Red Hat Enterprise Linux Server includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?

□ Software Development
□ Web server

You can further customize the software selection now, or after install via the software management application.

● Customize later ○ Customize now



Click next to begin installation of Red Hat Enterprise Linux Server.

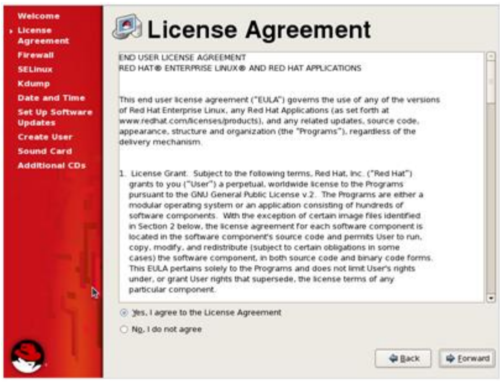
A complete log of the installation can be found in the file '/root/install.log' after rebooting your system.

A kickstart file containing the installation options selected can be found in the file '/root/anaconda-ks.cfg' after rebooting the system.

POST INSTALLATION CONFIGURATIONS (Not needed for exam)

1. Accept the Red Hat License agreement and click 'Forward.

2. The next step is regarding the configuration of the Firewall. You can Enable or Disable the firewall according to your preferences and then click 'Forward'.



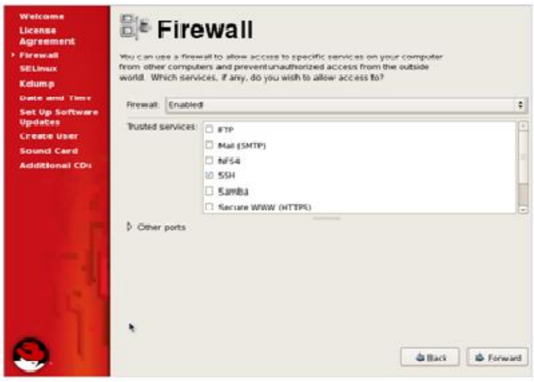
License Agreement

END USER LICENSE AGREEMENT
RED HAT® ENTERPRISE LINUX® AND RED HAT APPLICATIONS

This end user license agreement ("EULA") governs the use of any of the versions of Red Hat Enterprise Linux, any Red Hat Applications (as set forth at www.redhat.com/licenses/products), and any related updates, source code, appearance, structure and organization (the "Programs"), regardless of the delivery mechanism.

1. License Grant. Subject to the following terms, Red Hat, Inc. ("Red Hat") grants to you ("User") a perpetual, worldwide license to the Programs pursuant to the GNU General Public License v.2. The Programs are either a modular operating system or an application consisting of hundreds of software components. With the exception of certain image files identified in Section 2 below, the license agreement for each software component is located in the software component's source code and permits User to run, copy, modify, and redistribute (subject to certain obligations in some cases) the software component, in both source code and binary code forms. This EULA pertains solely to the Programs and does not limit User's rights under, or grant User rights that supersede, the license terms of any particular component.

● Yes, I agree to the License Agreement
○ No, I do not agree



Firewall

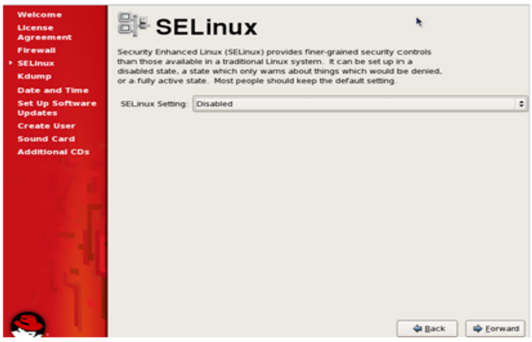
You can use a firewall to allow access to specific services on your computer from other computers and prevent unauthorized access from the outside world. Which services, if any, do you wish to allow access to?

Firewall: Enabled

Trusted services:
□ FTP
□ Mail (SMTP)
□ NFS4
☑ SSH
□ Samba
□ Secure WWW (HTTPS)

3. Next step is about the configuration of another security tool- SE Linux. By default you should keep it 'Disabled' unless you're working with very secure information.

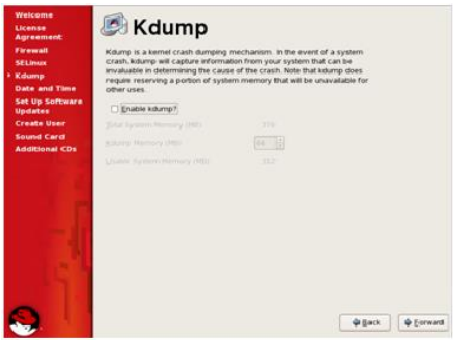
4. You can also choose to configure the Kdump that stores information about system crashes if your system fails but uses valuable disk space. By default don't enable it.



SELinux

Security Enhanced Linux (SELinux) provides finer-grained security controls than those available in a traditional Linux system. It can be set up in a disabled state, a state which only warns about things which would be denied, or a fully active state. Most people should keep the default setting.

SELinux Setting: Disabled



Kdump

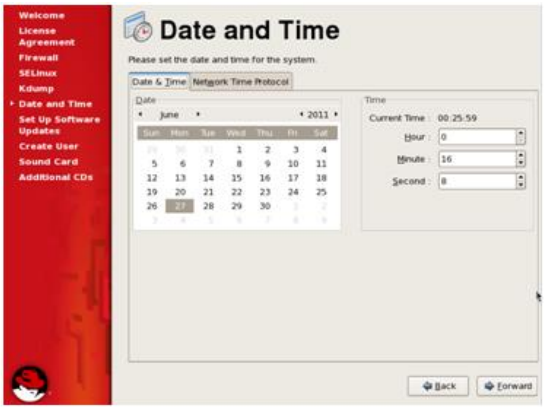
Kdump is a kernel crash dumping mechanism. In the event of a system crash, kdump will capture information from your system that can be invaluable in determining the cause of the crash. Note that kdump does require reserving a portion of system memory that will be unavailable for other uses.

□ Enable kdump?

Total System Memory (MB): 376
Kdump Memory (MB): 64
Usable System Memory (MB): 312

5. Next, adjust the time and date settings, and then click 'Forward'.

6. The next step is for software updates from Red Hat, at this point you should skip the registration and register at a later time and then click 'Forward'.



Date and Time

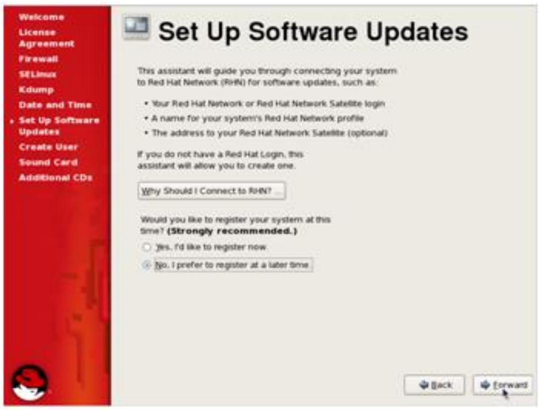
Please set the date and time for the system.

Date: June 2011

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	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	1	2
3	4	5	6	7	8	9

Time

Current Time: 00:25:59
Hour: 0
Minute: 16
Second: 8



Set Up Software Updates

This assistant will guide you through connecting your system to Red Hat Network (RHN) for software updates, such as:

- Your Red Hat Network or Red Hat Network Satellite login
- A name for your system's Red Hat Network profile
- The address to your Red Hat Network Satellite (optional)

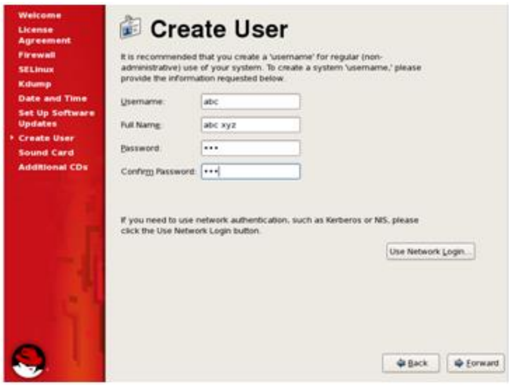
If you do not have a Red Hat Login, this assistant will allow you to create one.

Would you like to register your system at this time? **(Strongly recommended.)**

○ Yes, I'd like to register now
● No, I prefer to register at a later time

7. This step is to create a non-admin user for the system. Enter the details and click 'Forward'.

8. This step is about configuration of your Sound Card, choose a sound device and then click 'Forward'.



Create User

It is recommended that you create a 'username' for regular (non-administrative) use of your system. To create a system 'username,' please provide the information requested below.

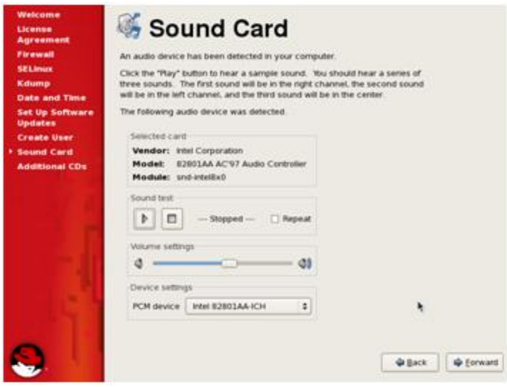
Username: abc

Full Name: abc xyz

Password: •••

Confirm Password: •••

If you need to use network authentication, such as Kerberos or NIS, please click the Use Network Login button.



Sound Card

An audio device has been detected in your computer.

Click the "Play" button to hear a sample sound. You should hear a series of three sounds. The first sound will be in the right channel, the second sound will be in the left channel, and the third sound will be in the center.

The following audio device was detected.

Selected card

Vendor: Intel Corporation

Model: 82801AA AC'97 Audio Controller

Module: snd-intel8x0

Device settings

PCM device: Intel 82801AA-ICH

9. Click 'Forward' to complete the configurations and start using your Red Hat Linux OS.
(Note- On clicking 'Finish' the system would require a reboot if you have made changes to the configurations of Firewall or SE Linux.)

Your Red Hat Linux OS is now all installed and configured, ready to be used. Good luck using and exploring various features of Red Hat Linux.

Partitioning of Hard Drives in Linux

While it is true that Linux will operate just fine on a disk with only one large partition defined, there are several advantages to partitioning your disk:
First, it may reduce the time required to perform file system checks because these checks can be done in parallel.
Second, with multiple partitions, you can, if you wish, mount one or more of your partitions as read-only.
Finally, the most important benefit that partitioning provides is protection of your file systems.
Personally, I favor the KISS principle (Keep It Simple, Stupid), especially for beginners. Desktop systems for personal use don't have most of the complications that require so many partitions. For a healthy Linux installation, I recommend three partitions: swap, root, and home.

Swap

You need one partition that will be used as Linux swap space. This is space on your hard drive that can be used as *virtual memory*. Virtual memory allows your computer to run large programs and perform complex tasks even if it does not have enough physical RAM to do the job.
As a rule of thumb, the swap partition should be at least double the amount of physical RAM installed in your system.

Root (/)

The *root* file system is represented by a forward slash (/). It is the top of the directory tree, and contains Linux and everything that you install with Linux.
This is roughly equivalent to you're "C:" drive under DOS or Windows.
The size of your root partition will vary depending on what you install or plan to install.
In recent version of Linux the file system of root partition is ext4.
In general, you should be fine with a root partition between 20GB and 40GB.

/home

This is the place where all the user-specific files, your data in other words, are stored. It is roughly equivalent to the "My Documents" folder on a MS Windows desktop. On a multi-user system, each user will have own directory under /home.
Strictly speaking, it is not necessary to create a separate partition for /home. If you do not, it will reside on the root partition like everything else. We can define /home partition space as our need.

Troubleshooting of installation

In the early years of Linux, it's hard to get Linux installation done on the first try. This is usually because of hardware compatibility and driver support.
A normal problem occurs during or after Linux installation would be the video card, network interface card (NIC), package installation problems etc. Other installation problems most certainly happen because user do not properly read or follow installation guide manual. The solution is very simple, start over, and this time please take it seriously, make proper preparations etc.
The best troubleshooting step is to perform a media check on your installation CD-ROMs. View system files and hardware to troubleshoot your Linux installation problem. Sometime the wrong partitioning can also raise the problem.
If you really cannot troubleshoot your Linux problems, you can search for solution in the Internet. Check out problem solution in your Linux distributions official forum. There are many Linux experts out there who are willing to help.

"Why let Microsoft give you Windows while Linux can give you a house?"

☺ ☺ ☺

IMPORTANT INTERVIEW QUESTIONS

1. Compare Linux & Windows.

Criteria	Linux	Windows
Type of OS	Open Source	Proprietary
Customization	High level of customization	Cannot be changed
Security	Excellent	Cannot be changed

2. What is LILO?

- LILO is Linux Loader is a boot loader for Linux. It is used to load Linux into the memory and start the Operating system. LILO can be configured to boot other operating systems as well. LILO is customizable, which means that if the default configuration is not correct, it can be changed. Config file for LILO is lilo.conf.
- LILO is also a code snippet which loads PC BIOS into the main memory at the time of starting the computer system. It handles the following tasks:
  -Locating Linux kernel
  -Identifying other supporting programs and loading them in the memory
  -Starting Kernel
- The selection of various kernel images and boot routines is supported by LILO. For this reason, it is known as boot manager.

3. You wish to print a file 'draft' with 60 lines on a page. What command would you use?

The command used: **pr -l60 draft**
– The default page length when using pr is 66 lines.
– The -l option specifies a different length.

4. What is Linux?

Linux is an operating system based on UNIX, and was first introduced by Linus Torvalds. It is based on the Linux Kernel, and can run on different hardware platforms manufactured by Intel, MIPS, HP, IBM, SPARC and Motorola. Another popular element in Linux is its mascot, a penguin figure named Tux.

5. What is the difference between UNIX and LINUX?

UNIX originally began as a propriety operating system from Bell Laboratories, which later on spawned into different commercial versions. On the other hand, Linux is free, open source and intended as a non-propriety operating system for the masses.

6. What is BASH?

BASH is short for Bourne Again SHell. It was written by Steve Bourne as a replacement to the original Bourne Shell (represented by /bin/sh). It combines all the features from the original version of Bourne Shell, plus additional functions to make it easier and more convenient to use. It has since been adapted as the default shell for most systems running Linux.

7. What is Linux Kernel?

The Linux Kernel is low-level systems software whose main role is to manage hardware resources for the user. It is also used to provide an interface for user-level interaction.

8. What is a swap space?

A swap space is a certain amount of space used by Linux to temporarily hold some programs that are running concurrently. This happens when RAM does not have enough memory to hold all programs that are executing.

9. What is the advantage of open source?

Open source allows you to distribute your software, including source codes freely to anyone who is interested. People would then be able to add features and even debug and correct errors that are in the source code. They can even make it run better, and then redistribute these enhanced source code freely again. This eventually benefits everyone in the community.

10. **What are the basic components of Linux?**
Just like any other typical operating system, Linux has all of these components: kernel, shells and GUIs, system utilities, and application program. What makes Linux advantageous over other operating system is that every aspect comes with additional features and all codes for these are downloadable for free.

11. **Does it help for a Linux system to have multiple desktop environments installed?**
In general, one desktop environment, like KDE or Gnome, is good enough to operate without issues. It's all a matter of preference for the user, although the system allows switching from one environment to another. Some programs will work on one environment and not work on the other, so it could also be considered a factor in selecting which environment to use.

12. **What is the basic difference between BASH and DOS?**
The key differences between the BASH and DOS console lies in 3 areas:
– BASH commands are case sensitive while DOS commands are not;
– under BASH, / character is a directory separator and \ acts as an escape character. Under DOS, / serves as a command argument delimiter and \ is the directory separator
– DOS follows a convention in naming files, which is 8 character file name followed by a dot and 3 character for the extension. BASH follows no such convention.

13. **What is the importance of the GNU project?**
This so-called free software movement allows several advantages, such as the freedom to run programs for any purpose and freedom to study and modify a program to your needs. It also allows you to redistribute copies of software to other people, as well as freedom to improve software and have it released to the public.

14. **Describe the root account.**
The root account is like a systems administrator account, and allows you full control of the system. Here you can create and maintain user accounts, assigning different permissions for each account. It is the default account every time you install Linux.

15. **What is CLI?**
CLI is short for Command Line Interface. This interface allows user to type declarative commands to instruct the computer to perform operations. CLI offers an advantage in that there is greater flexibility. However, other users who are already accustom with using GUI find it difficult to remember commands including attributes that come with it.

16. **What is GUI?**
GUI, or Graphical User Interface, makes use of images and icons that users click and manipulate as a way of communicating with the computer. Instead of having to remember and type commands, the use of graphical elements makes it easier to interact with the system, as well as adding more attraction through images, icons and colors.

17. **How do you open a command prompt when issuing a command?**
To open the default shell (which is where the command prompt can be found), press Ctrl-Alt-F1. This will provide a command line interface (CLI) from which you can run commands as needed.

18. **How can you find out how much memory Linux is using?**
From a command shell, use the "concatenate" command: cat /proc/meminfo for memory usage information. You should see a line starting something like: Mem: 64655360, etc. This is the total memory Linux thinks it has available to use.

19. **What is typical size for a swap partition under a Linux system?**

The preferred size for a swap partition is twice the amount of physical memory available on the system. If this is not possible, then the minimum size should be the same as the amount of memory installed.

20. **What are symbolic links?**
    Symbolic links act similarly to shortcuts in Windows. Such links point to programs, files or directories. It also allows you instant access to it without having to go directly to the entire pathname.

21. **Does the Ctrl+Alt+Del key combination work on Linux?**
    Yes, it does. Just like Windows, you can use this key combination to perform a system restart. One difference is that you won't be getting any confirmation message and therefore, reboot is immediate.

22. **How do you refer to the parallel port where devices such as printers are connected?**
    Whereas under Windows you refer to the parallel port as the LPT port, under Linux you refer to it as /dev/lp. LPT1, LPT2 and LPT3 would therefore be referred to as /dev/lp0, /dev/lp1, or /dev/lp2 under Linux.

23. **Are drives such as harddrive and floppy drives represented with drive letters?**
    No. In Linux, each drive and device has different designations. For example, floppy drives are referred to as /dev/fd0 and /dev/fd1. IDE/EIDE hard drives are referred to as /dev/hda, /dev/hdb, /dev/hdc, and so forth.

24. **How do you change permissions under Linux?**
    Assuming you are the system administrator or the owner of a file or directory, you can grant permission using the chmod command. Use + symbol to add permission or – symbol to deny permission, along with any of the following letters: u (user), g (group), o (others), a (all), r (read), w (write) and x (execute). For example the command chmod go+rw FILE1.TXT grants read and write access to the file FILE1.TXT, which is assigned to groups and others.

25. **In Linux, what names are assigned to the different serial ports?**
    Serial ports are identified as /dev/ttyS0 to /dev/ttyS7. These are the equivalent names of COM1 to COM8 in Windows.

26. **How do you access partitions under Linux?**
    Linux assigns numbers at the end of the drive identifier. For example, if the first IDE hard drive had three primary partitions, they would be named/numbered, /dev/hda1, /dev/hda2 and /dev/hda3.
27. **What are hard links?**
    Hard links point directly to the physical file on disk, and not on the path name. This means that if you rename or move the original file, the link will not break, since the link is for the file itself, not the path where the file is located.

28. **What is the maximum length for a filename under Linux?**
    Any filename can have a maximum of 255 characters. This limit does not include the path name, so therefore the entire pathname and filename could well exceed 255 characters.

29. **What are filenames that are preceded by a dot?**
    In general, filenames that are preceded by a dot are hidden files. These files can be configuration files that hold important data or setup info. Setting these files as hidden makes it less likely to be accidentally deleted.

30. **Explain virtual desktop.**
    This serves as an alternative to minimizing and maximizing different windows on the current desktop. Using virtual desktops, each desktop is a clean slate where you can open one or more programs. Rather than minimizing/restoring all those programs as needed, you can simply shuffle between virtual desktops with programs intact in each one.

31. How do you share a program across different virtual desktops under Linux?
To share a program across different virtual desktops, in the upper left-hand corner of a program window look for an icon that looks like a pushpin. Pressing this button will "pin" that application in place, making it appear in all virtual desktops, in the same position onscreen.

32. What does a nameless (empty) directory represent?
This empty directory name serves as the nameless base of the Linux file system. This serves as an attachment for all other directories, files, drives and devices.

33. What is the pwd command?
The pwd command is short for print working directory command. It's counterpart in DOS is the cd command, and is used to display the current location in the directory tree.

34. What are daemons?
Daemons are services that provide several functions that may not be available under the base operating system. Its main task is to listen for service request and at the same time to act on these requests. After the service is done, it is then disconnected and waits for further requests.

35. How do you switch from one desktop environment to another, such as switching from KDE to Gnome?
Assuming you have these two environments installed, just log out from the graphical interface. Then at the Log in screen, type your login ID and password and choose which session type you wish to load. This choice will remain your default until you change it to something else.

36. What are the kinds of permissions under Linux?
There are 3 kinds of permissions under Linux:
– Read: users may read the files or list the directory
– Write: users may write to the file of new files to the directory
– Execute: users may run the file or lookup a specific file within a directory

37. How does case sensitivity affect the way you use commands?
When we talk about case sensitivity, commands are considered identical only if every character is encoded as is, including lowercase and uppercase letters. This means that CD, cd and Cd are three different commands. Entering a command using uppercase letters, where it should be in lowercase, will produce different outputs.

38. What are environmental variables?
Environmental variables are global settings that control the shell's function as well as that of other Linux programs. Another common term for environmental variables is global shell variables.

39. What are the different modes when using vi editor?
There are 3 modes under vi:
– Command mode – this is the mode where you start in
– Edit mode – this is the mode that allows you to do text editing
– Ex mode – this is the mode wherein you interact with vi with instructions to process a file

40. Is it possible to use shortcut for a long pathname?
Yes, there is. A feature known as filename expansion allows you do this using the TAB key. For example, if you have a path named /home/iceman/assignments directory, you would type as follows: /ho[tab]/ice[tab]/assi[tab] . This, however, assumes that the path is unique, and that the shell you're using supports this feature.

41. What is redirection?
Redirection is the process of directing data from one output to another. It can also be used to direct an output as an input to another process.

42. **What is grep command?**
grep a search command that makes use of pattern-based searching. It makes use of options and parameters that is specified along the command line and applies this pattern into searching the required file output.

43. **What could possibly be the problem when a command that was issued gave a different result from the last time it was used?**
One highly possible reason for getting different results from what seems to be the same command has something to do with case sensitivity issues. Since Linux is case sensitive, a command that was previously used might have been entered in a different format from the present one. For example, to lists all files in the directory, you should type the command ls, and not LS. Typing LS would either result in an error message if there is no program by that exact name exist, or may produce a different output if there is a program named LS that performs another function.

44. **What are the contents in /usr/local?**
It contains locally installed files. This directory actually matters in environments where files are stored on the network. Specifically, locally-installed files go to /usr/local/bin, /usr/local/lib, etc.). Another application of this directory is that it is used for software packages installed from source, or software not officially shipped with the distribution.

45. **How do you terminate an ongoing process?**
Every process in the system is identified by a unique process id or pid. Use the kill command followed by the pid in order to terminate that process. To terminate all process at once, use kill 0.

46. **How do you insert comments in the command line prompt?**
Comments are created by typing the # symbol before the actual comment text. This tells the shell to completely ignore what follows. For example: "# This is just a comment that the shell will ignore."

47. **What is command grouping and how does it work?**
You can use parentheses to group commands. For example, if you want to send the current date and time along with the contents of a file named OUTPUT to a second file named MYDATES, you can apply command grouping as follows: (date cat OUTPUT) > MYDATES

48. **How can you append one file to another in Linux?**
To append one file to another in Linux you can use command **cat file2 >> file 1.** The operator >> appends the output of the named file or creates the file if it is not created. While another command **cat file 1 file 2 > file 3** appends two or more files to one.

49. **Explain how you can find a file using Terminal?**
To find a file you have to use command, **find. –name "process.txt"** . It will look for the current directory for a file called process.txt.

50. **Explain how you can create a folder using Terminal?**
To create a folder, you have to use command ***mkdir.*** It will be something like these : ~$ mkdir Guru99

51. **Explain how you can view the text file using Terminal?**
To view the text file, go to the specific folder where the text files are located by using the command **cd** and then type **less filename.txt.**

52. **Explain how to enable curl on Ubuntu LAMP stack?**
To enable curl on Ubuntu , first install libcurl, once done use following command **sudo/etc/init .d /apache2 restart** or **sudo service apache2 restart.**

53. **Explain how to enable root loging in Ubuntu?**
The command which enables root loging is
#sudo sh-c 'echo "greater-show-manual-login=true" >>/etc/lightdm/lightdm.conf'

54. How you can run an Linux program in the background simultaneously when you start your Linux Server?
By using **nohup.** It will stop the process receiving the **NOHUP** signal and thus terminating it you log out of the program which was invoked with. **&** runs the process in the background.

55. Name a service that you should disable (which acts both as Web and FTP Server) on a Linux Server.
The **finger service** should be disabled because a remote user can get important information about the system by using that command.

56. What does Sar provide? Where are Sar logs stored?
Sar collects, reports, or saves system activity information, sar serves to log and evaluate a variety of information regarding system activity. With performance problems, sar also permits retroactive analysis of the load values for various sub-systems (CPUs, memory, disks, interrupts, network interfaces and so forth) and limitation of problems in this manner. If CPU utilization is near 100 % (user + nice + system), the workload sampled is CPU-bound.

57. How can you enhance the security of password file?
Linux keep user account information in a text file called /etc/passwd. This file also stores one way encrypted password. It is accessed by several tools to get user information, which is a security risk, so file need to 'Word Readable.' To minimize the security risk, you can use shadow password format. This method saves account information in regular file /etc/passwd. However, the password is stored as a single "x" character (not actually stored in this file). A second file, called "/etc/shadow" contains encrypted password as well as other information such as account or password expiration values, etc. The /etc/shadow file is readable only by the root account and is therefore less of a security risk

58. Suppose your FTP Server is hacked and the entire server needs to be restored. How should you restore the original kernel system files?
You cannot restore the entire operating system from tape backup device. So you should reinstall the core operating system, and then restore system configuration files and user data from tape backup device.

59. Why should you avoid Telnet to administer a Linux system remotely?
Telnet uses most insecure method for communication. It sends data across the network in plain text format and anybody can easily find out the password using the network tool. In the case of Telnet, these include the passing of login credentials in plain text, which means anyone running a sniffer on your network can find the information he needs to take control of a device in a few seconds by eavesdropping on a Telnet login session.

60. What is Puppet Server?
Puppet is open-source & enterprise software for configuration management toll in UNIX like operating system. Puppet is an IT automation software that is used to push configuration to its clients (puppet agents) using code. Puppet code can do multiple tasks from installing new software to checking file permissions to updating user accounts.

61. What command is used to check the number of files, disk space and each user's defined quota?
repquota command is used to check the status of the user's quota along with the disk space and number of files used.
This command gives a summary of the user's quota that how much space and files are left for the user. Each user has a defined quota in Linux. This is done mainly for the security as some users have only limited access to files. This provides a security to the files from unwanted access. The quota can be given to a single user or to a group of users.

62. Can Linux computer be made a router so that several machines may share a single Internet connection? How?
Yes, a Linux machine can be made a router. This is called "IP Masquerade." IP Masquerade is a networking function in Linux similar to the one-to-many (1: Many) NAT (Network Address Translation) servers found in

many commercial firewalls and network routers. The IP Masquerade feature allows other “internal” computers connected to this Linux box (via PPP, Ethernet, etc.) to also reach the Internet as well. Linux IP Masquerading allows this functionality even if the internal computers do not have IP addresses.

63. **How shadow passwords are given by in Linux?**
pwconv command is used for giving shadow passwords. Shadow passwords are given for better system security. The pwconv command creates the file /etc/shadow and changes all passwords to ‘x’ in the /etc/passwd file. First, entries in the shadowed file which don’t exist in the main file are removed. Then, shadowed entries which don’t have `x’ as the password in the main file are updated. Any missing shadowed entries are added. Finally, passwords in the main file are replaced with `x’. These programs can be used for initial conversion as well to update the shadowed file if the main file is edited by hand.

64. **What is YUM?**
YUM stands for Yellow dog Updater, Modified because it is based on YUP, the Yellow dog Updater. Where does the name Yellow dog come from? Yellow Dog is a version of Linux for the Power Architecture hardware and is RPM-based, just like Red Hat Enterprise Linux and Fedora. YUP, and later YUM, were written by the Linux community as a way to maintain an RPM-based system.

65. **What is the role of “Kudzu”?**
Kudzu is used to detect new Hardware. RedHat Linux runs a hardware discoverer, named kudzu. When attempting to identify a serial port Kudzu resets the serial port. This stops the serial console. Kudzu is configured from the file/etc/sysconfig/kudzu. Kudzu can be prevented from resetting hardware, by setting the configuration parameter SAFE to yes.

66. **What is the difference between ext2 and ext3 file systems?**
    - The ext3 file system is an enhanced version of the ext2 file system.
    - The most important difference between Ext2 and Ext3 is that Ext3 supports journaling.
    - After an unexpected power failure or system crash (also called an unclean system shutdown), each mounted ext2 file system on the machine must be checked for consistency by the e2fsck program. This is a time-consuming process and during this time, any data on the volumes is unreachable.
    - The journaling provided by the ext3 file system means that this sort of file system check is no longer necessary after an unclean system shutdown. The only time a consistency check occurs using ext3 is in certain rare hardware failure cases, such as hard drive failures.

67. **Explain /proc filesystem?**
/proc is a virtual file system that provides detailed information about Linux kernel, hardware and running processes. Files under /proc directory named as Virtual files.
Since /proc contains virtual files, it is called **virtual file system**. These virtual files have unique qualities. Most of them are listed as zero bytes in size.
Virtual files such as /proc/interrupts, /proc/meminfo, /proc/mounts, and /proc/partitions provide an up-to-the-moment glimpse of the system’s hardware. Others: /proc/filesystems file and the /proc/sys/ directory provide system configuration information and interfaces.

68. **What Squid service does?**
Squid provides the Proxy service, which caches and filters web traffic.

69. **Which popular office suite freely available for both Microsoft and Linux ?**
Open office suite is freely available. You can install it on both OS Microsoft and Linux.

70. **You company recently switched from Microsoft to Linux. All their documents are created on Microsoft office 2007. How will you manage documents created on Microsoft office 2007?**
Install open office suite in Linux. It allows you work with Microsoft documents.

71. **What is the SMTP?**
SMTP is the most common protocol for an e-mail server.

72. **What Samba service does?**
The Samba service provides Microsoft SMB support in Linux so that Linux machines can connect to Microsoft network resources.

73. **Name any of three spreadsheet programs, those are equivalent to Microsoft Excel and available for Linux?**
Openoffice calc [Free], KOffice Kspread [Free], StarOffice Base [commercial], Corel Quattro [commercial]

74. **Name a connection-less protocol for file transfer.**
TFTP is a connection-less protocol for file transfer.

75. **You company wants to run Web Server on their intranet. Which Linux package should you use for this?**
To run a Web server, you should install the Apache Web server.

76. **You company have slow internet connection. Which Linux service you can use to manage internet connection?**
You should use Squid proxy server, which allows managing the web contents and also caching the web pages to decrease the amount of traffic going to Internet.

77. **Which IP address should you use for client desktop?**
For client you can use both static and dynamic method. For easy management using DHCP to assign IP address would be the best option.

78. **Android which uses modified Linux Kernel for mobile devices has so much security flaws, why?**
Well Android is developed in Java Programming Language and Java itself is known to have a number of security flaws. Moreover Android is very much in its child stage and will take some time to get matured.

79. **Is Linux Operating System Immune to Malware?**
To be true, **No**! No **OS** on this earth can be ever be **100%** immune to **Viruses** and **Malware**. But still **Linux** never had a widespread **malware-infection** as compared to **Windows. Linux** is architecturally strong and hence very much immune (not totally) to security threats. **Linux** is **Kernel** and **GNU/Linux** is the **OS**. There are hundreds of distributions of **Linux**.
Now suppose a malicious script is written for a **RPM** based system i.e., **RedHa**t, **Fedora**, **CentOs**, it can't infect a **Debian** based system and a destructive script written for **Debian** based **OS** can't infect a **RPM** based System. Moreover a script that is going to perform a system-wide change needs **root** password. If **root** password is confidential and strong enough, the **OS** is literally secure. Now a windows virus cannot contaminate a **Linux** until Wine is installed and run as root. Hence it is suggested not to run wine as **root**.
Theoretically, a computer virus, in its most basic sense, is a program and we can run programs on Linux.

80. **How secured is Linux? Explain.**
Security is the most important aspect of an operating system. Due to its unique authentication module, Linux is considered as more secured than other operating systems. Linux consists of PAM. PAM is Pluggable Authentication Modules. It provides a layer between applications and actual authentication mechanism. It is a library of loadable modules which are called by the application for authentication. It also allows the administrator to control when a user can log in. All PAM applications are configured in the directory "/etc/pam.d" or in a file "/etc/pam.conf". PAM is controlled using the configuration file or the configuration directory.

81. **What is the minimum number of partitions you need to install Linux?**
Minimum 2 partitions are needed for installing Linux. The one is / or root which contains all the files and the other is swap. Linux file system is function specific which means that files and folders are organized according to their functionality. For example, all executable are in one folder, all devices in another, all libraries in another and so on. / or 'root' is the base of this file system. All the other folders are under this one. / can be consider as C: .Swap is a partition that will be used as virtual memory. If there is no more

available RAM a Linux computer will use an area of the hard disk, called swap, to temporarily store data. In other words it is a way of expanding your computers RAM.

82. What are the partitions created on the mail server hard drive?

The main partitions are done firstly which are root, swap and boot partition. But for the mail server three different partitions are also done which are as follows:
1. /var/spool- This is done so that if something goes wrong with the mail server or spool than the output cannot overrun the file system.
2. /tmp- putting this on its own partition prevents any user item or software from overrunning the system files.
3. /home- putting this on its own is useful for system upgrades or reinstalls. It allow not to wipe off the /home hierarchy along with other areas.

83. How do you create a new user account?

useradd command is used for creating a new user account. When invoked without the -D option, the useradd command creates a new user account using the values specified on the command line and the default values from the system. The new user account will be entered into the system files as needed, and initial files copied, depending on the command line options. This command uses the system default as home directory. If –m option is given then the home directory is made.

84. Which password package is installed for the security of central password?

Shadow password packages are used for security of central passwords. Security is the most important aspect of every operating system. When this package is not installed the user information including passwords is stored in the /etc/passwd file. The password is stored in an encoded format. These encoded forms can be easily identified by the System crackers by randomly encoding the passwords from dictionaries. The Shadow Package solves the problem by relocating the passwords to another file (usually /etc/shadow). The /etc/shadow file is set so that it cannot be read by just anyone. Only root will be able to read and write to the /etc/shadow file.

85. When do you need a virtual hosting?

The term Virtual Host refers to the practice of maintaining more than one server on one machine, as differentiated by their apparent hostname. For example, it is often desirable for companies sharing a web server to have their own domains, with web servers accessible as www.company1.com and www.company2.com, without requiring the user to know any extra path information.

86. Where the kernel modules are located?

The '/lib/modules/kernel-version/' directory stores all kernel modules or compiled drivers in Linux operating system. Also with 'lsmod' command we can see all the installed kernel modules.

87. What is umask?

Umask stands for 'User file creation mask', which determines the settings of a mask that controls which file permissions are set for files and directories when they are created.

88. How to change the default run level in linux?

To change the run level we have to edit the file "/etc/inittab" and change initdefault entry (id:5:initdefault:). Using 'init' command we change the run level temporary like 'init 3' , this command will move the system in runlevl 3.

89. What is Network Bonding?

Network bonding is the aggregation of multiple Lan cards into a single bonded interface to provide fault tolerance and high performance. Network bonding is also known as NIC Teaming.

90. How to upgrade Kernel in Linux?

We should never upgrade Linux Kernel , always install the new New kernel using rpm command because upgrading a kenel can make your linux box in a unbootable state.

91. How to check the ip address of LAN Card?
Using 'ifconfig' & 'ip address' command we can determine the ip address of LAN Card.

92. How To check the default gatway?
Using 'route -n' command we can determine the default gateway in linux.

93. Which Command is used to check the kernel Version?
'uname -r'

94. What is Initrd?
Initrd stands for initial ram disk , which contains the temporary root filesystem and neccessary modules which helps in mounting the real root filesystem in read mode only.

95. What does firewall do?
Firewall is a device or service which can be used to protect the network or system from other outside the networks.

96. During installation Linux creates a swap space partition. Why do we need this and how is it different from a Windows swap file?
Like Windows, Linux uses a certain amount of space for holding programs temporarily, when there is not enough available RAM to hold all the programs that are running concurrently. Generally, the least recently used program (or part of a program) is copied from memory to a file on your hard drive until it is needed again, at which time the current least recently used program is swapped out in its place and the first program is loaded back into memory. This file is called a swap file in Windows and "swap space" in Linux, but in either case it is a form of data file that is read from and written to off and on as long as your system is running. Windows puts the swap file (a hidden system file with different names for different versions of Windows) in the bootable data partition by default. Linux, by default, requires a special swap partition in which to store the swap file.

97. How large a swap partition should I create?
Although it can be smaller, for best results the partition size should be at least equal to the amount of memory installed in the system—preferably twice the amount of physical memory. In other words, if you have 64MB of RAM installed use double that amount for your swap space partition (64MB * 2 = 128MB).

98. I have a printer attached to the parallel port. What is this port called in Linux?
When prompted for the name of a parallel port in Linux, in place of LPT1, LPT2, or LPT3 use /dev/lp0, /dev/lp1, or /dev/lp2, respectively.

99. Is there a way to increase the priority of a program to make it run faster?
Yes. If you have a program that normally takes quite a while to run and you really, really need it to complete faster, you can boost its job priority level so that it uses more processor cycles. Of course, this means that all other currently executing programs will run slower. But if this is what you want, the nice command will do the job for you. Then, to start a program called Myprog with a high priority-level of –10, use the command: nice –n –10 myprog.

100. What is a virtual desktop?
When running a number of programs concurrently, the screen sometimes becomes cluttered with several windows open at once, covering each other up. One solution, of course, is to minimize some of the windows until they are needed. However, this entails minimizing and restoring those windows repeatedly, as different windows are accessed. This can be tedious over the course of a day.
An alternative is to use a program that creates "virtual desktops." Each desktop is a clean slate where you can open one or more programs. Rather than minimizing/restoring all those programs as needed, you can simply shuffle between virtual desktops with programs intact in each one.

101. The ls command doesn't show everything in the directory. What's wrong?

The ls (lowercase L, not capital i) utility has many parameters to let you view the directory contents in different ways. For example, ls –a shows all files and ls –al shows all files in "long" form (with additional details), while ls –a --color adds a splash of color to differentiate the various types of files. (For the list of parameters, use: ls --help or type: man ls for a full description.)

102. What is "chmod" command? What do you understand by this line "r-- -w- --x?

chmod command is used to change permission of a file or directory in UNIX. The line you see shows the permission for three different set of people : user, group and others.

User is the currently logged in user, while group is for all other member which are part of certain group and others means anyone other than user and group member. Each group has three permissions rwx stands for read, write and execute and they are written as user_group_others. So in above line, user has only read permission, group members have write permissions and other people have only execute permission. If it is a directory then you need execute permission to go inside that directory.

103. What is Freeware?

Freeware is software that is distributed without demanding a fee for its usage. These programs are available either as fully functional software for an unlimited period. Ownership of any freeware is retained by its developer. The developer can change future releases from freeware to a paid product (freeware) if he wishes so. Also, a freeware is typically distributed without its source code.

104. What is Crippleware?

Some software are offered as freeware – but with very limited features – or with the major feature missing. These are refered to as Crippleware.

105. What is Free Software?

Well, free software is software that gives a user freedom to run, copy, distribute, study, change and improve software. To be precise, free software is a matter of liberty, not price!

And unlike freeware, free software may be distributed for a fee. Please note, to modify a program you need to access its source code which free software offers whereas a freeware does not.

106. What is Open-Source Software?

The term 'Open Source' is very close to 'free software' but not identical to it. We say this because, the source code of open-source software is readily available to users too but under a copyright, and one is freely allowed to re-distribute the software.

107. What is Shareware?

Shareware is demonstration software that is distributed for free but for a specific evaluation period only, say, 15-30 days (**Trialware**). After the evaluation period the program gets expired and a user can no longer access the program.

108. What is Adware?

Adware, better known as advertising software is software that automatically renders advertisements. Most of these advertisements appear in the form of annoying pop-ups. However, one can disable the ads by purchasing a registration key. It can even change your home page, default search or install a tool bar. Like freeware, *Adware too* is available for computer users at no cost.

THANK YOU . . .

"In a world of no walls or fences, we don't need WINDOWS or GATES."