Chapter 1

Objective

- Explaining the nature of open-source software
- Discuss the origins of Linux
- List the Red Hat operating system distributions
- Explain basic Linux principles

Introduction to Red Hat Enterprise Linux

- Enterprise-targeted operating system
- Focused on mature open source technology
- 18-24 month release cycle
  - Certified with leading OEM and ISV products
- Two variants available
  - Server
  - Client
- Purchased with one year Red Hat Network subscription and support contract
  - Support available for seven years after release
  - Up to 24X7 coverage plans available

Redhat Network

- A comprehensive software delivery, system management, and monitoring framework
  - Update Module: Provides software updates
  - Management Module: extended capabilities for large developments
  - Provisioning Module (bare-metal installation, configuration management, and multi-stage configuration rollback capabilities)
  - Monitoring Module

Fedora Project

- Red Hat sponsored open source project
- Focused on latest open source technology
  - Rapid four to six month release cycle
  - Available as free download from the Internet
- An open, community-supported proving ground for the technologies which may be used in upcoming enterprise products
- Red Hat does not provide formal support

Open source

- Open source: software and source code available to all
  - The freedom to distribute software and source code
  - The ability to modify and create derived works
  - Integrity of author's code
- The Free software foundation and the Four Freedoms
  - The software and the source code must be freely distributable.
  - All must be able to modify the source code and create derived works
  - To maintain the integrity of the original author's work, the license may require that changes to the code be provided in patch form.
- Four Freedoms
  - The software must be freely executable for any purpose.
  - The source code must be available so that other can study how it works
  - The software must be freely redistributable.
  - All are free to modify the software

Linux Origins

- 1984: The GNU Project and the free software Foundation
  - Creates open source version of UNIX utilities
  - Creates the General Public License (GPL)

    - Software license enforcing open source principles
- 1991: Linux Torvalds
    - Creates open source, UNIX-like kernel, released under the GPL
- Now a days:
    - Linux kernel+GNU utilities=complete, open source, UNIX like operating system

Redhat Distributions are OSes based on the Linux kernel

- Red Hat Enterprise Linux
    - Stable, thoroughly tested software
    - Professional support services
    - Centralized management tools for large networks
- The Fedora Project
    - More, newer applications
    - Community supported(no official Red Hat support)
    - For personal systems

Other Linux Distributions,

- NepaLinux: desktop version localized in Nepali language
- **Slackware**, one of the first Linux distributions, founded in 1993, and since then actively maintained by Patrick J. Volkerding
- **Debian**, a non-commercial distribution maintained by a volunteer developer community with a strong commitment to free software principles
- **Ubuntu**, a newly popular desktop distribution maintained by Canonical that is derived from Debian
- **CentOS**, a distribution derived from the same sources used by Red Hat, maintained by a dedicated volunteer community of developers with both 100% Red Hat - compatible versions and an upgraded version that is not always 100% upstream compatible.
- **Mandriva**, a Red Hat derivative popular in France and Brazil, today maintained by the French company of the same name
- **openSUSE**, originally derived from Slackware, sponsored by the company Novell
- **Gentoo**, a distribution targeted at power users, known for its FreeBSD Ports-like automated system for compiling applications from source code
- Knoppix, a LiveCD distribution that runs completely from removable media and without installation to a hard disk
- **Linspire**, a commercial desktop distribution based on Ubuntu (and thus Debian), and once the defendant in the Microsoft vs. Lindows lawsuit over its former name.
- Portable Linux
    - Puppy Linux
- Other free distribution apart from Linux
    - FreeBSD
    - Open Solaris

(Reference: http://en.wikipedia.org/wiki/Linux_distribution)

Basic Unix/Linux principles

- **Every thing is a file**
    - It helps to control the access in hardware by treating them as file.
- **Small, single-purpose programs**
    - Small program to perform one task very well so that to avoid complexity in program.
- **Ability to chain programs together to perform complex tasks**
    - Core design feature of UNIX is that the output of one program can be the input for another. This gives the user the flexibility to combine many small programs together to perform a larger, more complex task.
- **Avoid captive user interfaces**

    - Interactive commands are rare in UNIX. Most commands expect their options and arguments to be typed on the command line when the command is launched. It may be helpful when automating task without needing human.
- **Configuration data stored in text**
    - Clear text based configuration file which can also be manipulated with simple text editor.
    - Configuration file can be easily moved from one system to another.

Reference:
Course book from Redhat Enterprise Linux

Chapter 2

Objective

- Log into a Red Hat Enterprise Linux system
- Start X from a console
- Access the command line from X
- Change your password
- Understand the nature of root privileges
- Elevate your privileges
- Edit plain text files

Log in to a Linux system

- Two type of logins
  - Virtual Consoles (text base) [6 console available by default]
  - Graphical logins (called display managers)
- Login using login name and password
- Each user has a home directory for personal file storage

Logging in

- In consol mode press ctrl+Alt+{F1 to F6} for different virtual consol (text base consol). If you are in one console and want to move to another console you can also use Alt+{F1 to F6}
- To log into graphical mode press Ctrl+Alt+F7

Element of the X Window System

- The X window system is Linux's graphical subsystem
- Xorg is the particular version of the X window System used by Red Hat
- Look and behavior largely controlled by the desktop environment.
- Two most widely used desktop are
  - GNOME: the default desktop environment
  - KDE: an alternate desktop environment provided by Redhat.

Starting the X server

- X server (GUI) may or may not start at boot time.
- If X don't start at boot time we can start it by using
  #startx
  #X
  Or
  #init 5
  By logging into the virtual console.

The root user

- The root user: a special administrative account
  - Also called the super user
  - Root has near complete control over the system
- Do not login as root unless necessary (in real life).

Changing Identities

- su – creates new shell as root
- sudo command runs command as root (if configured by root for it)
- id shows information on the current user

Editing text files

- The nano editor
  - Easy to learn, easy to use
  - Not as feature-packed as some advanced editors
- vi editor, an advanced full feature editor
- other editors

- gedit
- pico (old version of nano)
- gvim

Chapter 3

Objective

- Execute commands at the prompt
- Explain the purpose and usage of some simple commands
- Use the built-in help resources in Red Hat Enterprise Linux

Running Commands

- Commands have the following syntax:
  - Command options arguments
- Each item is separated by a space
- Options modify a command's behavior
  - Single-letter options usually preceded by –
    # ls -l
  - Full-word options usually preceded by --
    #ls --help
  - S
- Arguments are filenames or other data needed by the command.
- Multiple commands can be separated by ;
- Press Ctrl+c to interrupt running command

Examples

$ mkdir backups; cd backups ; pwd
Note:
mkdir: is used to make directory
cd: is used to change directory location
pwd: provides the location of working directory.
$date
Tue Dec 4 12:34:56:04 EDT 2002
$date +"Today is %A, %B %d, %y.%nIt is %r, %Z."
Today is Tuesday, December 04, 2007.
It is 12:35:41 PM, EST.

$cal 9 2007
September 2007

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	1	12	1	1	15
		1		3	4	
16	17	1	19	2	2	22
		8		0	1	
23	24	2	26	2	2	29
		5		7	8	
30	31	3	33	3	3	36
		2		4	5	
37	38	3	40	4	4	43
		9		1	2	

Getting Help

- It is not possible to memorize all the commands
- Many level of helps are available
  - whatis
  - command –help
  - man and info
  - /usr/shared/doc/

- Red Hat Documentation

Example

$whatis netstat

$ifconfig --help

$man arp

Note:

Printed by –help, man and other helps, used to describe the syntax of command:

- Arguments in [ ] are optional
- Arguments in CAPS or <> are variables
- Text followed by … represents a lost
- x|y|z means "x or y or z"
- -abc means "any mix of -a, -b, -c"

The man command

- Provides documentation for commands
- Almost every command has a man page
- Pages are grouped into chapters
- Man [<chapter>] <command>
- Example
  - #man 1 passwd
  - #man 5 passwd
- While viewing a man page
  - Navigate with arrow, PgUP, PgDn
  - /text searches for text
  - n/N goes to next/previous match
  - q quits
- Searching the Manual
  - Man –k keyword lists all matching pages
  - Uses whatis database

The info command

- Similar to man, but often more in-depth
- Run info without args to list all page
- Info pages are structured like a web site
  - Each page is divided into "nodes"
  - Links to nodes are preceded by *
  - Info [command]
- While viewing an info page
  - Navigate with arrows, PgUP, PgDn
  - Tab moves to next link
  - Enter follows the selected link
  - n/p/u goes to the next/previous/up-on node
  - s text searches for text (default: last search)
  - q quits info

Extended Documentation

- The /usr/share/doc directory contain the documentation of installed program.
- Subdirectories for most installed packages
- Location of docs that do not fit elsewhere
  - Example configuration files
  - HTML/PDF/PS documentation
  - License details

Red Hat documentation

- Available in web http://www.redhat.com/docs/

Unit 4

Browsing the File system
Objective

- Understanding importance elements of the file system hierarchy
- Copy, move, and remove files
- Create and view files
- Manage files with Nautilus

Linux File Hierarchy Concepts

- Files and directories are organized into a single-rooted inverted tree structure
- File system begins at the root directory, represented by a lone / character
- Names are case-sensitive
- Paths are delimited by /
- Each shell and process on the system has designated current or working directory.
- .. refers to thee parent directory of any particular directory –one level up in the file hierarchy.
- . refers to the current directory.
- Files and directories whose names begin with a . are hidden
- A user's path is a list of directories that are searched for commands typed at the command line.

Some important Directories

- Home Directories /root for root user, /home/username for other users
- User Executables: /bin, /usr/bin, /usr/local/bin
- System Executables: /sbin, /usr/sbin, /usr/local/sbin
- Other mounts: /media, /mnt
- Configuration: /etc
- Temporary Files: /tmp
- Kernels and Bootloader: /boot
- Server Data: /var, /srv
- System Information: /proc, /sys
- Shared Libraries: /lib, /usr/lib, /usr/local/lib

Current Working Directory

- Each shell and system process has a current working directory (cwd)
- Pwd
  - Displays the absolute path to the shell's cwd

File and Directory Names

- Names may be up to 255 characters
- All characters are valid, except the forward-slash
  - It may be unwise to use certain special
  - Some characters should be protected with quotes when referencing them
- Name are case-sensitive
  - Example: Mail, MAIL, mail, and mAiL
- To access a file whose name contains special characters, enclose the filename in quotes. For example
  - #ls –l "file name.txt"

Absolute and relative Pathnames

- **Absolute pathnames**
  - Begin with a forward slash
  - Complete "road map" to file location
  - Can be used anytime you wish to specify a file name
- **Relative pathnames**
  - Do not begin with a slash
  - Specify location relative to your current working directory
  - Can be used as a shorter way to specify a file name

Example:
#mkdir parentA
#mkdir parentB
#cd parentA
#touch fileInA
#touch ../parentB/fileInB
#cp ../parentB/fileInB .

Changing directory

- cd changes directories
  - To an absolute or relative path:
    - #cd /home/xyz/
    - #cd project/docs
    - To a director one level up:
    - #cd ..
  - To you home directory
    - #cd

    Or
    - #cd ~
    - ~ sign indicate home directory
  - To your previous working directory:
    - #cd –

Listing Directory Contents

- Lists the contents of the current directory or a specified directory
- Usage:
  - Ls [options] [file_or_dir]
- Examples:
  - #ls –a (include hidden files)
  - #ls –l (display extra information)
  - #ls –R (recurs through directories)
  - #ls –ld (directory and symlink information)

Copying Files and Directories

- cp – copy files and directories
- **Usage:**
  - Cp [options] file destination
  - More than one file may be copied at a time if the destination is a directory:
  - #cp [options] file1 file2 dest
- **Important options**
  - -i (interactive) ask before overwriting a file
  - -r (recursive): recursively copy an entire directory tree
  - -p (preserve): preserve permissions, ownership, and time stamps
  - -a (archive) copies files and directories recursively (like –r) while preserving permission (like –p)
- **Example**
  - #cp ~root/myfile /tmp
  - #ls /tmp
  - Here ~root represents home of root user
- **Copying files and Directories: The destination**
  - If the destination is a directory, the copy is placed there
  - If the destination is a file, the copy overwrites the destination.
  - If the destination does not exist, the copy is renamed
- **To copy directory with content:**
  - #cp –r mydirectory /mnt
  - The directory mydirectory will be copied into /mnt directory
- **Moving and Renaming Files and directories**
  - mv – moves and/or rename files and directories
  - Usage:
    - Mv [options] file destionation
  - More than one file

- Moving files and Directories: The destination
  - If the destination is a directory, the file/directore is placed there
  - If the destination is not a directory(directory with same named doesn't exist) the file is renamed.
  - Example:
  - #mkdir parentA
  - #touch file1
  - #move file1 parentB
    - It will rename file1 to parentB
  - #move parentB parentA
    - It will move the file parentB into directoryB

- **Creating and Removing Files**
  - touch filename : creates empty files or update file timestamps
  - rm –remove files
  - Usage:
    - rm [options] <file> ..
  - Example
    - #rm –I filename (interactive)
    - #rm –r directory (recursive: to remove directory with contents)
    - #rm –f file (force)
    - Rm –rf directory (combining recursive and force options)
- Create and Remove Directories
- mkdir: creates directory
- rmdir removes empty directory
- rm –r remove the non empty directory.

  Determining File content
- file : display the type of file
- **Example**
  - #file a.html
  - Will show HTML document text
- For more detail see /usr/share/magic