Unit 1 Introduction to Linux

What is an Operating System?

An operating system is made of software instructions that lie between computer hardware and the application programs .At the centre is the KERNEL, which provides the basic computing functions.

A Brief History of UNIX

The root of Unix lie in CTSS (Comprehensive Time Sharing System) developed by F.Cobarto at MIT in early 1960, continuing the same principles of CTSS a multi-user and multi-user system was designed under the Multiplexed Information Computing Service(MULTICS) project started by General Electrics and AT&T. Later in 1970 a system called Uniplexed Information and Computing Service (UNICS) was developed at Bell Labs.

What is Linux?

Linux is a free operating system was created by Linus Torvalds in 1991 when he was a student at University of Helsinki. The Linux operating system was created on POSIX standards (Portable Operating System Interface for UNIX).In 2003 Red Hat changed the name of it's distribution from Red hat Linux to Fedora Core and moved it commercial efforts toward RHEL products, It then setup Fedora to be

Sponsored by Red Hat
Supported by Linux community
6-12 month lifecycle.

Common Linux Features

- Multi-user
- Multitasking
- Hardware Support
- GUI (X Window system)
- Networking connectivity
- Applications support

Linux Variants

-
- Debian
- SUSE (Novell)
- Mandarake
- Slackware
- Gentoo
- Ubuntu

UNIX Variants

- HP-UX
- Solaris
- Irix (Silicon Graphics)
- AIX

RHEL Features

- Ideal for for Enterprise environment.
- 16-24 months of lifecycle
- One year subscription must be purchased for support and software update.
- Designed in 2 forms

  * Server
  * Desktop

- The most popular Redhat supported software.

  * GFS - Used in Cluster File systems.
  * Jboss – Middleware Applications
  * Directory server - Similar to Active Directory based on LDAP
  * Certificate server – Identity Management
  * Redhat Application stack - Includes JBOSS , MySQL / POSTREGSql / Apache

- It supports 19 languages.

What is Open Source?

- The software and the source code must be feely distributable.
- All must be able to modify the source code
- To maintain the integrity of the original work the license may require that changes to the code be provided in patch form
- The license must be inherited
- The license must be nondiscriminatory

Unit 2 – Linux Usage Basics

Logging in to a Linux system.

The default gui mode is gnome (gnu network object model environment), Optionally you can install the KDE - K desktop environment

Linux system can be accessed via either command mode / graphical mode

Command mode is also known as Virtual Console.

There 6 virtual consoles available , in order to toggle between the consoles you must type

CTRL + ALT + F1 **(F1** to **F6** to move in to 6 consoles)

To switch from the GUI to Virtual Console.

Method 1

CTRL + ALT + F7

Method 2

Open a terminal windows and type

[root@labpc ~]#**init 3**

To switch from the Virtual Console to GUI

Method 1

[root@labpc ~]#startx

Method 2

[root@labpc ~]#init 5

Method 3

CTRL + ALT + F7

Changing your password

From the GUI , Click on

System -> Preference -> About Me -> Change Password

NOTE :- Follow the screen message to enter and confirm the password. It is a best practice use a complex password.

From the Terminal

[student@labpc ~]$passwd

Explanation of prompt

"[student@labpc ~]$"

student – Is the name of the user account who logged in

labpc - Is the hostname

~ - Is the name of the current working directory

$ - Is the privilege level $ means non root.

For example the root user who is working on the /etc directory will have the following prompt.

"[root@labpc etc]#"

Unit 3 – Running Commands and Getting help.

Commands give instruction to your server , some of the commands need to be run with the root user permission.
Commands can run one at a time or at the same time. Some of the commands are given below, each commands and it' argument must be separated with a space.

- To print the system date

  #date
  Mon Oct 10 09:25:02 AST 2011

- To set the system date

  #date MMDDHHSSYY – To set the date to November 12 2011 10:00 AM

  # date 1112100011

- To print the Calendar

  #cal

- When you are on root user and to change user "student" password

  #passwd student

  NOTE- Only root can change the password of another user .

- To check your login session in the system

#id

To run multiple commands at one for eg:- To get the calendar and the current date on on the system.

#cal;date

The rule of thumb is we should never memories the commands , The linux system provides several commands and documents to get help

* whatis
* command --help
* man and info
* /usr/share/doc
* Redhat documentation.

The whatis command has the following characteristics.

- display short description of the commands
- use a database that updates nightly
- Often not available after the install immediately.

$whatis history

```
history              (3)  - GNU History Library
history [builtins]   (1)  - bash built-in commands, see bash(1)
```

The numbers shown here is the page number in the linux manual page.

The --help option

- display usage summary and argument list
- Used by most not all commands.

$cat --help

```
Usage: cat [OPTION] [FILE]...
Concatenate FILE(s), or standard input, to standard output.
  ----------some output below omitted----------
```

* Anything in straight braces [] is optional
* Anything followed by .... represents an arbitrary -length list of that thing
* If you see multiple options separated by pipes | it means you can use any one of them
* Text in <> brackets represents variable data , So <filename> means "insert the filename you wish to use here.

The man command

- Provides documentation for commands
- Almost of every commands has a man page
- Pages are grouped to chapters

$ man 3 history ----> show s the third chapter of the command history.

To scroll between use the PageUp and the PageDown Key or Up/Down Arrow

To search a text phrase type " /text " you want to search , in order to move to the next matched phrase press "n"

To quit from the MAN page press "q"

To search the man pages using a keyword for eg:- To search all the commands which has the word "disk"

$man -k disk

The info command

- similar to man but often more in-depth
- run info without arguments to list all pages

#info history

Unit 4 – Browsing the File Systems

Linux file hierarchy concepts

* Files and directories are organized into a single -rooted inverted tree structure
* Filesystem begins at the root "/ "
* Names are case sensitive
* Paths are delimited by "/"
* .. refers to the parent directory of any particular directory.
* . refers to the current directory
* Files and directories begin with "." are hidden
* Directory and file names must be max 255 characters.

Some Important Directories

/home - every user has a home directory all of the user's personal files data go here , Root users home directory is /root , and the non root home directories usually named after the user.

/bin - The essential binaries or executables

/media – All the removable media is loaded to the filesystem is mounted in to a sub directory of /media , a cdrom will mount in /media/cdrom.whenever if you want to read a file from the cdrom you must access /media/cdrom directory.

/etc - most of the configuration files are stored here.

/boot - The boot loader , kernel and loader configuration files are stored here.

/var - Regularly changing files such as logs , print spools and e-mail pools are stored here

/proc - Provided information about a running linux system and allows some tweaking while system is running

/tmp - is used by application to store it's temporary data.

Print your current working directory

[student@labpc log]$ ***pwd***
/var/log

To list the directory contents

[student@labpc etc]$ ***ls***

To list the directory contents in human readable format

[student@labpc etc]$ ***ls -lh***
drwxr-xr-x 2 root root 4.0K Apr 19 13:27 pm
drwx------ 2 root root 4.0K Jan 21 2009 ppp

1st columns shows the file type such letter "d" is for a directory "-" is for a file
rwx – stands for read write execute permission , so first 3 rwx for the user or owner of the object
And the second for the groups and 3rd for the others
.
2nd columns shows the the no of objects

3rd & 4th columns shows the owner and the group of the objects.

5th columns the size of the file

6th created date time

7th name of the object.

To list the directory contents recursively (It will display folder, files and folder contents)

[student@labpc etc]$ ***ls -lR***

To list the hidden directory and files

[student@labpc etc]$ ***ls -la***

There are two way where a path can be identified in linux

* relative path
To list the contents of your current directory /home/student you can enter

[student@labpc etc]$ ls -l ./

* absolute path
eg: - same as above

[student@labpc etc]$ ***ls -l /home/student.***

To change the directory

[student@labpc etc]$ cd /home/admin

To change it to your home directory

[student@labpc etc]$ ***cd***

OR

[student@labpc etc]$ ***cd ~***

To change it to your previous working directory

[student@labpc etc]$ ***cd -***

To Copy a file

[student@labpc etc]$ ***cp sourcefile targetfile***

[student@labpc etc]$ ***cp /home/student/testfile /tmp/***
Will copy the testfile to the tmp directory

[student@labpc etc]$ ***cp /home/student/testfile /tmp/test1file***
Will copy the testfile to the tmp directory with the a different name

[student@labpc etc]$ ***cp -p /home/student/testfile /tmp/***
Will copy the testfile to the tmp directory without changing the timestamps and permissions.

To Move a file

[student@labpc etc]$ ***mv sourcefile targetfile***

student@labpc etc]$ ***mv /home/student/testfile /tmp/***
Will move the testfile to the tmp directory

student@labpc etc]$ mv ***/home/student/testfile ./changelog***
Will rename the testfile to changelog in the current directory

To remove a file

[student@labpc etc]$ rm -i ***testfile***
Will prompt you with a message asking for the confirmation , never use the "rm" command ,but when the necessity arises use it with "-i" option

[student@labpc etc]$ ***rm -rf /tmp***
Will remove the non empty folders and files recursively.

To remove a directory

[student@labpc etc]$ ***rmdir /tmp***
Will remove /tmp diretory if it is not empty.

To create a directory

[student@labpc etc]$ ***mkdir tmp***
Will create a directory tmp in the current location

To create a file

[*student@labpc etc]$* ***touch testfile***
Will create a file testfile in the current location.
When the command "touch" is executed on an existing file it will update the timestamps

To identify the file type

[*student@labpc etc]$* ***file /var/log/messages***
/var/log/messages: ASCII text ----→ for text files

[*student@labpc etc]$* ***file /usr/bin/passwd***
/usr/bin/passwd: setuid ELF 32-bit LSB executable ----→ for executable.

NOTE:- The above steps can be done via the GUI using the nautilus to access it
Application -> System Tools -> FileBrowser

Unit 5 – Users , Groups and Permissions.

Users

* Every user has a unique user ID (UID) – By default root user has UID "0"
* Username and passwords are stored in ***/etc/passwd***
* Users are assigned a home directory and a program that runs when they login.
* Users cannot read or write each other files with out proper permissions.

Groups

* Users are assigned to groups
* Each group has it's own Group ID
* GroupID are stored in /etc/group

Linux File Security

* Every file/object is owned by a UID and GID
* Every process runs unded UID and GID
* Three access categories
 - User
 - Group
- Others

Permission Types

* Read (r)
* Write (w)
* Executable (x)
* None (-)

To check the current permission

*[student@labpc etc]$**ls -l /home/student/test***
-rw-r--r-- 1 student root 1805 Jun 26 10:22 //home/student/test

student - User - Read & Write
root - Group – Read Only
other - Others - None

To change file/ folder ownership

*[root@labpc ~]#**chown –R student1 testfile***
Will change the owner of the file "testfile" to the user "student1"

Following 2 command will produce the same results.

chown user:group file

chown user file;chgrp group file

To change file/ folder group membership

*[root@labpc ~]#**chgrp –R students testfile***
Will change the group for the file "testfile" to the group "students"

Changing permission in symbolic method

*[root@labpc ~]#**chmod –R mode file***

modes are –
u , g , o -> for users , group , others
+ , - -> for grant or deny
r , w ,x -> for read , write , executable

*[root@labpc ~]#**chmod –R u+rw,g-r file***
Will grant the user read,write & the read permissions will be removed for the group

Changing permission in numeric method

4 – read
2 - write
1 – executable

*[root@labpc ~]#**chmod –R 750 file***
User will have the -> read , write , executable rights
Group wil have the -> read & executable rights
Other wil have no permissions

NOTE : The above procedures can be performed via the FileBrowser in GUI

Unit 6 – Using the bash shell

File Globbing -- > Is known as wildcard representation, by using wildcard expression we can avoid any repetitive entries in our commands.

* - matches zero or more characters
? - matches any single characters
[0-9] – matches a range of numbers
[abc] – matches any of the character in the list
^a - first letler a.

[:keyword:] → alpha , upper , lower , digit , alnum , punct , space

*[root@labpc ~]#**rm *.mp3***

*[root@labpc ~]#**echo ?e****

*[root@labpc ~]#**echo [^[:upper:]]****

Retreiving the previous entered commands

*[root@labpc ~]#**history***

** ctrl +r – provide a search*
** alt +. - repeat the commands*

Autocompleting the word

By pressing the TAB key you can complete a command or an argument.

Brace Expansion or $

*[root@labpc ~]#**mkdir -p car/parts/{door,engine,mirror}***
Will make the folder as below
car -> parts -> door
engine
mirror

*[root@labpc ~]#**echo "myname is $hostname"***
Will print the text -→ my name is student

Command editing

To edit the previously entered command we could use the following key.

CTRL + *a* - > move the cursor beginning of the line
CTRL + *e* -> move the cursor to the end of line
CTRL + *k* -> deletes to end of line
CTRL + *u* -> deletes to beginning of the line.

Bash Scripting basics

Bash scripting is a process of grouping a collection of commands to automate the taks, rather than typing the command frequently.

* step 1 - Use a text editor to create a file containing the commands.
1st line should be #!/bin/bash

* step 2 - enter the commands to you need to run

* step 3 - save the file

* step 4 – make it executable
chmod +x filename.

* step 5 - ./filename - > to run the bash script.

Sample script

```
#!/bin/bash
# This script displays some information about your environment
echo "Greetings. The date and time are $(date)"
echo "Your working directory is: $(pwd)"
```

Unit 7 – Standard I/O & Pipes

Standard Input and Output

* Linux provides three I/O channels to Programs
* Standard input (STDIN) - keyboard by default
* Standard output (STDOUT) - terminal window by default
* Standard error (STDERR) - terminal window by default

STDIN - Value of the file descriptor is 0
STDOUT – Value of the file descriptor 1
STDERR - Valuse of the file descriptor 2

We could redirect the output and the error to a file

Redirecting the standard output to a file

*[student@labpc ~]$**find /etc –name passwd > outputfile***

Redirecting the standard error and output to a file

*[student@labpc ~]$**find /etc –name passwd 2> outputfile***

Ignoring or discarding the standard error

If we are not worried about the error we could use /dev/null which will empty the file (equivelant to Recycle bin)

*[student@labpc ~]$**find /etc –name passwd > outputfile 2>/dev/null***

*[student@labpc ~]$**find /etc –name passwd > /dev/null 2>&1*** → *This will hide STDOUT & STDERR*

NOTE :- To append we could use the " >> "

Redirecting the standard output to a program

In order to redirect a command output to another command input we could use the | sign.

*[student@labpc ~]$**ls –l | wc –l***
111 → *This command will give the line number of the directory listing rather than the directory output.*

Redirecting all the output (standard & error) to a file.

*[student@labpc ~]$**find /etc –name passwd &> outputfile** → "&" means all"*

Redirecting to multiple targets

*[root@labpc ~]#**ls -lR /etc | tee log1 |sort –r |tee log2***
The command "tee" will bes useful for saving output at various stages in a long sequence of pipes.

Sending Multiple lines to STDIN

You could type multipel lines via STDIN until a particular word is sent"

[root@labpc ~]#mail -s "Test Message" root <<END →

```
> Test 2
> Test 3
> END
```

When you type the word END it will close the session , This can be any word as per your requirement.

Unit 8 – Text Processing Tools

Tools to read text

*[root@labpc ~]#**cat /etc/passwd -> whole page at one***

*[root@labpc ~]#**less /etc/passwd -> can scroll up/down***

*[root@labpc ~]#**head /etc/passwd -> display first 10 lines (use "-number oflines" to display the required number of lines)***

*[root@labpc ~]#**tail /etc/passwd -> display last 10 lines (use "-number oflines" to display the required number of lines)***

Tools to extract text

*[root@labpc ~]#**grep student /etc/passwd***
Will print the line containing student.

*[root@labpc ~]#**grep -v student /etc/passwd***
Will print all the lines except the containing student.

*[root@labpc ~]#**cut -f3 -d:/etc/passwd***
** -d - delimeter*
**-f - the column number*

Tools for analyzing t text

*[root@labpc ~]#**wc /etc/passwd -> will give you the number of lines , word and the bytes***

-l → for line count
-w →for word count

*[root@labpc ~]#**sort /etc/passwd →sort the passwd file***

-n →numeric sort
-r →reverse sort

*[root@labpc ~]#**cut –d: -f7 /etc/passwd | uniq →will print only the unique line.***

Comparing files and patching the difference.

*[root@labpc ~]#**diff -u file1 file 2 > file.patch***
Will find the difference on file 1 and the out put is stored on the file.patch

*[root@labpc ~]#**patch –b file1 file.patch***
This will change the differed data.

Checking spelling

*[root@labpc ~]#**aspell check testfile***
Will open an interactive spell checking tool.

Translating Altering & the text

*[root@labpc ~]#**ls -l | tr 'a-z' 'A-Z'*** *- > will translate the listing from lowercase to uppercase.*

*[root@labpc ~]#**sed 's/date/time/' test*** → *will change the word date to time in the file test.*

Unit 9 – Advance Text Editor – VI

This is a powerful tool we used to create or modify config files in linux system.Anybody who Wish to become a linux expert must get familiar with this powerful tool

[root@labpc ~]#vi testfile -> Will open a new file called testfile for editing.

3 main modes

**Insert mode*
**Command mode*
**Ex mode*

NOTE : - To move between the mode press ESC , for eg:- While editing , if you want to save the file , press ESC then give the :wq!

Below are list of options available in VI editor for editing

To append - press a
To insert - press i
To insert a line below – press 0
To insert a line above – Press O
To change the uppercase or lowercase – press ~
To copy a line -yy
To copy multiple lines (for eg:- 4 lines) – Type 4yy
To cut a line – Type dd
To paste a line – p
To insert line number – set number
To disable or hide the line numbet – set nonumber
To move to a particular line number – 5G – will take the cursor to 5th line
To move the cursor to particular word "apple" - /apple press Enter
To delete the trailing line from the current cursor point - d$
To replace a single character - Press r
To replace a word – Type Shift + r
To delete a single character – press X
To undo changes – u
To redo the actions - .
To save the file - :w
To save & exit the file - :wq!
To exit without saving – q!
To navigate through the file - j – move the cursor left
h – move the cursor right
k - move the cursor up
j - move the cursor down.

Unit 10 – Basic System Configuration Tools

TCP/IP Network Configuration

Network interfaces are named sequentially: eth0, eth1, etc
* Multiple addresses can be assigned to a device with aliases
* Aliases are labeled eth0:1, eth0:2, etc.
* Aliases are treated like separate interfaces

*[root@labpc ~]**#ifconfig eth0***
Will display the network configurations for the interface eth0

*[root@labpc ~]**#ifup eth0***
Will enable the interface

*[root@labpc ~]**#ifdown eth0***
Will disable the interface

Configuring the network

Device configuration are stored in
`/etc/sysconfig/network-scripts/ifcfg-ethX`

For detailed guide on all the options on configuration date refer the below
`/usr/share/doc/initscripts-*/sysconfig.txt`

`Sample configuration`

DHCP

```
DEVICE=ethX
HWADDR=0:02:8A:A6:30:45
BOOTPROTO=dhcp
ONBOOT=yes
Type=Ethernet
```

Manual

```
DEVICE=ethX
HWADDR=0:02:8A:A6:30:45
BOOTPROTO=static
IPADDR=192.168.0.254
NETMASK=255.255.255.0
GATEWAY=192.168.2.254
ONBOOT=yes
Type=Ethernet
```

Configuring the Hostname.

*[root@labpc ~]#**vi /etc/sysconfig/network***

```
NETWORKING=yes
HOSTNAME=student.lab.com
```

Configuring the DNS

[root@labpc ~]#vi /etc/resolv.conf

nameserver 192.168.0.254

Unit 11– Investigating and Managing Process

Whatis a Process

A process is an executing program with several components and properties, including a memory context, priority, and
environment. The Linux kernel tracks every aspect of a process by its PID under /proc/PID.

Viewing the process

*[root@labpc ~]#**ps –au***

** a – will display all the process*
** u - will display the user information*

Obtaining the Process ID

*[root@labpc ~]#**pgrep rpcbind -- > will provide the process ID***

OR pidof -→ will need the exact process name.

OR pstree → will give the process tree

Signalling the process

Signals are messages send to a certain process for an action such to stop a process

* Signal 15 – Gives a clean termination (TERM or SIGTERM)
* Signal 9 - Terminate immediately.
*Signal 1 - reload the configuration files.

*[root@labpc ~]#**kill -15 rpcbind***

Setting Priority to a Process

Scheduling priority determines access to the CPU
* Priority is affected by a process' nice value
* Values range from -20 to 19 but default to 0
* Lower nice value means higher CPU priority

*[root@labpc ~]#**nice -n 5 myprocess*** → *start a new process with priority*

*[root@labpc ~]#**renice 15 -p PID*** → *restart an existing process*

Interactively managing process

*[root@labpc ~]#**top***

Job Control

*[root@labpc ~]#vi **testprocess*** → *Will push the process to the background*
CTRL + Z

OR
*[root@labpc ~]#**testprocess &***

[root@labpc ~]#jobs → *will list the jobs running in the background*

[root@labpc ~]#bg [JOB NUMBER] → *will resume the job in the background*

[root@labpc ~]#fg [JOB NUMBER] → *will resume the job in the foreground.*

Scheduling Jobs using crontab

Entry consists of five space-delimited fields followed by a command line
* One entry per line, no limit to line length
* Fields are minute, hour, day of month, month, and day of week
* Comment lines begin with #
* See man 5 crontab for details

Minute 0-59
Hour 0-23
Day of Month 0-31
Month 0-12
Day of Week 0-6 (0=Sunday)
@daily /@hourly

*[root@labpc ~]#**crontab –e*** → *Will edit the crontab*

```
0 4 * * 1,3,5 mail -s Test  test@test.com
```

*[root@labpc ~]**#crontab -l*** → *Will list the crontab*

Exit Status

Processes report success or failure with an exit status
* 0 for success, 1-255 for failure
* $? stores the exit status of the most recent command

*[root@labpc ~]**#echo $?*** → *When we run this command after the process it will display the exit status*

*[root@labpc ~]**#exit 3*** → *To manually set a exit status – used in batch programming.*

Conditinal Execution Paramenters

Commands can be run conditionally based on exit status
* && represents conditional AND THEN
* || represents conditional OR ELSE

*[root@labpc ~]**#ping -c2 1.1.1.1 && echo "Test Message "***

*[root@labpc ~]**#ping -c 1.1.1. || echo "Test Message"***

1st command wont display the echo message , whereas the 2nd command does.

Unit 12– Managing the BASH Shell

Setting a variable

These are used in Shell scripting or advanced programming.

*[root@labpc ~]#**HI=Hello**. → You should not leave a space before and after the "=".*
*[root@labpc ~]#**echo $HI***

Environment Variables

** Variables are local to a single shell by default*
** Environment variables are inherited by child shells*
** Set with export VARIABLE=VALUE*
** Accessed by some programs for configuration*

*[root@labpc ~]#**env** → To check the Environment variables.*

*[root@labpc ~]#**export HISTSIZE=5** → will set the history buffer to 10*

Setting Aliases

It is always to difficult to type long commands , inorder to overcome this taedious issue we could set aliases

*[root@labpc ~]#**alias listdir='ls –l'** → Will create an alias called listdir*

[root@labpc ~]#listdir → Next time when we want to list the directory we could run the alias listdir

Tips

Below are few shell manipulation techniques

*[root@labpc ~]#*echo my cost is - \$5 → try with and without slash.

[root@labpc ~]echo ' * * Tips * * *' → Try with and without the quote.*

Unit 13– Finding and Processing Files

find [directory...] [criteria...]
* Searches directory trees in real-time
* Slower but more accurate than locate
* All files are matched if no criteria given
* Can execute commands on found files
* May only search directories where the user has read and execute permission

*[root@labpc ~]**#find / -name apple* → *Find the files with the name apple*

*[root@labpc ~]**#find/ -name test* -ok rm {} \;***
*Will remove all the files matching the word test and will prompt your before eexcuting the rm command for each file , Instead of **-ok** if you use **–exec** the command will run with out interaction.*

find / -type d → Will find only the directories.

some useful find options

* -o → to use as OR
* -not →
* -uid → user ID
* -perm → User permissions.
*-size → size + or –

Unit 14– Network Clients

GUI Web Clients
* Firefox

Non GUI browsers
*[root@labpc ~]#**links http://www/google.com***

wget
*[root@labpc ~]#**wget http://www.google.com/htp.exe***
Will download the htp.exe to your current working directory.

OpenSSH: Secure Remote Shell
* Secure replacement for older remote-access tools
* Allows authenticated, encrypted access to remote systems
* ssh [user@]hostname
* ssh [user@]hostname command

*[root@labpc ~]#**ssh -l student 1.1.1.1*** → *To connect to machine 1.1.1.1 via ssh*

*[root@labpc ~]#**ssh –l student 1.1.1.1 df -h*** → *This will check the disk space.*

SCP
scp source destination

*[root@labpc ~]#**scp testfile 1.1.1.1:/home/student*** → ***Will copy the testfile to the remote host in /hom/student directory.***

smbclient

*[root@labpc ~]#**smbclient -L 1.1.1.1*** → *List all the shares in 1.1.1.1*

*[root@labpc ~]#**smblient –U student //1.1.1.1/test*** → *Will connect to the share "test"*

-W - → *For workgroup / domain name.*

xterm

*[root@labpc ~]#ssh **-X -l student 1.1.1.1 xterm***
Will open a X window via the command with encryption , users can invoke GUI based commands

Network Diagnostic Tools
- *ping*
- *traceroute*
- *host*
- *dig*
- *netstat*

Unit 15– Advance Topics in Users , Group & Permissions

User and Group Information Files
When a user runs a command such as ls -l that displays user and group information about files, the numeric information is translated into names; it is the names that are displayed. The mappings of numbers to names are in the files /etc/passwd and /etc/group. The /etc/shadow file maps user names to their encrypted passwords and password and account expiration information. All files are colon separated.

The /etc/passwd file contains seven fields: user name, password placeholder (for historical reasons), uid number, gid number of the user's primary group, GECOS field (typically containing the user's real name), home directory, and shell to be run when a user logs in.

The /etc/group file contains four fields: group name, group password placeholder, gid number, and a comma separated list of group members.

The /etc/shadow file is referenced when someone logs in: the file contains a mapping of a user name to a password. For a complete list of the fields, see the man page:
man 5 shadow

System users and groups all have uid and gid numbers between 1 and 499. This excerpt from /etc/passwd shows several system users:

User Managemnet Tools

Graphical Tool

*[root@labpc ~]#**system-config-users***

Commandline Tools

[root@labpc ~]#useradd –d /home/jack –m jack
This command will create a user "jack"

[root@labpc ~]#userdel –r jack
Will delete the user jack and his home directories , if you skip "r" will retain the home directories.

[root@labpc ~]#usermod –g 520 jack
Will modify the group for the user Jack

Monitoring Logins

To identify the currenlty loggedin users and the process they are running..

*[root@labpc ~]#**w***
08:33:54 up 12 days, 20:16, 6 users, load average: 0.42, 0.35, 0.34
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root tty1 - 08Oct11 7days 0.16s 0.16s -bash
root :0 - 03Oct11 ?xdm? 9:48m 0.14s /bin/sh /usr/bin/startkde

*[root@labpc ~]#**last***
Will display the list of recently logged in users and the time of system reboots

*[root@labpc ~]#**lastb***
Will display the failed login attempts.

Default Permissions

* Default permission for directories is 777 minus umask
* umask is set with the **umask** command.
* Non-privileged users' umask is 002
  - Files will have permissions of 664
  - Directories will have permissions of 775
* root's umask is 022

*[root@labpc ~]#**umask*** → *Will show the default umask value*

*[root@labpc ~]#**umask 0222*** → *Will set an umask value*

Special Permissions for Executables

* Special permissions for executables:
* suid: command run with permissions of the owner of the command, not executor of the command
* sgid: command runs with group affiliation of the group of the command

The suid and sgid permissions are effective for executable regular files; the sticky bit and the sgid permission are effective for directories.
To set the special permissions, use the chmod command, preceding the usual three digits with a digit representing the special permission or permissions that you wish to have set:

4 for suid
2 for sgid
1 for the sticky bit

*[root@labpc ~]**#chmod 3777 testfolder*** → *Will set sgid and the stickybit for the folder.*

drwxrwsr-t 5 root root 4096 Oct 12 12:14 testfolder

Unit 16 - The Linux Filesystem In-Depth

Partitions and Filesystems

* Disk drives are divided into partitions
* Partitions are formatted with filesystems, allowing users to store data
* Default filesystem: ext3, the Third Extended Linux Filesystem
* Other common filesystems:
* ext2 and msdos (typically used for floppies)
* iso9660 (typically used for CDs)
* GFS and GFS2 (typically for SANs)

Inodes

* An inode table contains a list of all files in an ext2 or ext3 filesystem
* An inode (index node) is an entry in the table, containing information about a file (the metadata), including:
  - file type, permissions, UID, GID
  - the link count (count of path names pointing to this file)
  - the file's size and various time stamps
  - pointers to the file's data blocks on disk
  - other data about the file

[root@labpc ~]#ls -il → *will display the inode details*

```
80788 -rw-r----- 1 student student 5120 Sep 18 11:26 myData
37777 drwxr-x--- 2 student student 4096 Sep 18 11:25 newStuff
```

*[root@labpc ~]#**cp test ./myfolder/test*** → *Will create an inode as long it is in the same file system*

*[root@labpc ~]#**mv test /myfolder/test*** → *Will not make a change in the inode*

*[root@labpc ~]#**rm test*** → *will remove the inode entry.*

Hardlinks

*[root@labpc ~]#**ln fedora redhat***
*[root@labpc ~]#**ls -li fedora redhat***

```
246575 -rw-rw-r-- 2 digby digby 26 Sep 25 20:56 fedora
246575 -rw-rw-r-- 2 digby digby 26 Sep 25 20:56 redhat
```

Soft/ Symbolic links

*[root@labpc ~]# **ln -s /etc/passwd password***
*[root@labpc ~]# **ls -li password /etc/passwd***

```
30338 -rw-r--r-- 1 root root 1729 Aug 24 11:43 /etc/passwd
33276 lrwxrwxrwx 1 digby digby 11 Sep 26 09:33 passwd -> /etc/passwd
```

The Seven Fundamental Filetypes

- `-` regular file
- `d` directory
- `l` symbolic link
- `b` block special file
- `c` character special file
- `p` named pipe
- `s` socket

For c & b type os file refer the /dev directory.

p named pipe: a file that passes data between processes. It stores no data itself, but passes data between one process writing data into the named pipe and another process reading data from the named pipe. A named pipe can be created using the mknod command:

[root@labpc ~]# ***mknod mypipe p***

s socket: a stylized mechanism for inter-process communications. It is extremely rare for a user or even a system administrator to explicitly create a socket.

Checking Free Space

* df - Reports disk space usage
  - Reports total kilobytes, kilobytes used, kilobytes free per file system
  - -h and -H display sizes in easier to read units
* du - Reports disk space usage
* Reports kilobytes used per directory
* Includes subtotals for each subdirectory
* -s option only reports single directory summary

*[root@labpc ~]#**df -h*** → *Will display the free space*

*[root@labpc ~]#**du -h*** → *Will display the used space*

*[root@labpc ~]#**df -sh*** → *Will display the summary*

Removable Media

* Mounting means making a foreign filesystem look like part of the main tree.
* Before accessing, media must be mounted
* Before removing, media must be unmounted
* By default, non-root users may only mount certain devices (cd,dvd, floppy, usb, etc)
* Mountpoints are usually under /media

*[root@labpc ~]#**mkdir /media/cdrom***
*[root@labpc ~]#**mount /dev/sr0 /media/cdrom***

You could perform the same commands to mount the USB drives

*[root@labpc ~]#**mount /dev/sda1 /media/usb***
*[root@labpc ~]#**mount /dev/sr0 /media/cdrom***

*To find out the device name (sda1 / sr0) use the **dmesg** command.*

*[root@labpc ~]#**unmount /media/cdrom*** → *Will eject the cdrom*

Creating, Listing, and Extracting File Archives

tar

*[root@labpc ~]# **`tar -cvf /tmp/etc.tar /etc`***
Will create a tar file.

*[root@labpc ~]#**`tar -tf /tmp/etc.tar | less`***
Will list the contents of the compressed file.

*[root@labpc ~]#**tar -xvf /tmp/etc.tar***
To extract the contents.

-z will create a gzip archive
-j will create a bzip2 archive

zip & unzip

*[root@labpc **~]#zip etc.zip /etc***

*[root@labpc **~]#unzip etc.zip***

Unit 17 - Essential System Administration Tools

Managing Services

Service is an application running in the background waiting for request or performing tasks

chkconfig
[root@labpc ***~]#chkconfig --list sshd***
`sshd 0:off 1:off 2:on 3:on 4:on 5:on 6:off`
Will check the run status for each runlevel.

[root@labpc ***~]#chkconfig off sshd*** → ***Will disable the service in the startup***

[root@labpc ***~]#chkconfig on sshd*** → ***Will enable the service in the startup***

service
[root@labpc ***~]#service sshd status*** → *To check the status*
Instead of status you could use ***start , stop , restart***

Managing Software

* Software is provided as RPM packages
  - Easy installation and removal
  - Software information stored in a local database
* Packages are provided by Red Hat Network
  - Centralized management of multiple systems
  - Easy retrieval of errata packages
  - Systems must be registered first
  - Custom package repositories may also be used

Yum
* Front-end to rpm,
* Configuration in /etc/yum.conf and /etc/yum.repos.d/
* Used to install, remove and list software

[root@labpc ***~]# yum install packagename***
[root@labpc ***~]# yum remove packagename***
[root@labpc ***~]#yum update packagename***
[root@labpc ***~]#yum list available***
[root@labpc ***~]#yum list installed***

We could use the **yum** command to mange packages via rhn.

RPM

This is the Redhat Package Manager.

[root@labpc **~]#** rpm –ivh sendmail-8.13.8-8.el5

RPM Packages fromat is *name-version-release.architecture*.rpm

Managing Security

Selinux

* Kernel-level security system
* All processes and files have a context
* SELinux Policy dictates how processes and files may interact based on context
  - Policy rules cannot be overridden
  - Default policy does not apply to all services

Packet Filtering

* Network traffic is divided into packets
* Each packet has source/destination data
* Firewalls selectively block packets

Unit 18 – Discussion / Q & A