FIREBRAND

Linux

LPIC Level 1 Certification

LPIC Labs

Version 2.0

www.firebrandtraining.com

Linux

LPIC 1

Labs

Lab Setup
The computer you are using for lab exercises is configured as follows:

A host operating system running Fedora 15
User account is student and the admin account is root.
Both have the password of password
The host is configured with QEMU virtual machine manager
There is one virtual machines installed:
Debian 6

This machine has been installed and preconfigured ready for the labs to run.

Some of the labs will require the use of Fedora 21 so the first lab takes you through the process of installing F21 as a virtual machine.
The password for all accounts in all labs should be **password.** Please use this password for any accounts you create.

Lab - Installation of Fedora 21

First we need to create the virtual machine
With the Fedora 15 machine running go to **Applications, System Tools, Virtual Machine Manager.**
Click on the Icon to **Create a new virtual machine**.
Give the new machine a name of **F21** then click Forward
Select **Use ISO Image** and browse to the FC21 image by selecting Browse Local.
Use the image stored in **/home/student/Fedora21.iso**,
OS type **Generic**
Version **Generic**
Click **Forward**
Allocate **1024Mb** of memory and **1** CPU
Create a **Virtual Hard Disk of 10GB**, Select **Forward**
Leave other settings then Select **Forward**,
The hard disk will be created
The system should then boot the Fedora iso file
Select **Install to Hard disk**
Select **English** as the language and click **Continue** (Bottom right)
Select **UK** Keyboard type and click **Next**
Select **Installation Destination** for the disk then select **I will config partitioning**
Click **Done** (Top left)

Disk Partitioning
Click on the **+**
Configure the following

Mount Point - /boot
Standard Partition
File system type – EXT4
Size – 200mb

Click **+** again to add more partitions to continue.
Repeat this process for the two more partitions of the following types and sizes

Mount Point - /
File system type – EXT4
Size – 8G
Standard Partition

Mount Point - /home
File system type – EXT4
Size – 1G
Standard Partition

Click **Done**
Click **Done** again
If all correct **Accept Changes**
Click on **Begin Installation**

Set the **root password** to password
No need to setup any users at this stage

Once complete, click **Quit**

Now **reboot** the new vm by clicking on the **power icon** on the top right.

Once rebooted and you will be prompted with a series of Welcome questions

Lanuage: **English**
Typing**: English (UK)**
Timezone**: London**
No need to connect online
About you: **student**
Password: **password**

Finally click on **Start using Fedora**.

You can close the Getting started window or browse around.

.

Chapter 1
Lab - Text Processing

Task 1 - Echo pipes redirects and quotes

On your F15 machine login as student with the student password.
Copy the files from the /home/student/lpidocs/textproc directory into your home directory

```
$cd
$cp –R lpidocs/textproc ~
```

Redirection exercise

Type in the following in a shell

```
$echo “My name is $USER and my home directory is $HOME” > simple_echo
$cat simple_echo
```

We will now append to the file using >>

```
$echo ‘My Salary is $100’ >> simple _echo
$cat simple_echo
```

Notice how with single quotes the $ value is interpreted as a literal value but with double quotes the variable was expanded.

```
$cat simple_echo > new_echo
```

Notice how this copies a file. Same as using cp simple_echo new_echo

If we cat a file that doesn’t exist

```
$cat nofile
```

We get an error message. This is stderr. Let’s handle that using the 2>

```
$cat nofile 2> error_out
```

If you view the error_out it now contains the error messages that would have gone to the screen

We can send errors to the same place as stdout by using the 2>&1

```
$cat simple_echo nofile > allout 2>&1
```

We should now have a file that contains the simple_echo text and the error message in the file **allout**

Now let’s see what the << does. Type in the following

```
$cat << foobar
```

Hello foobar
foobar

Notice how it only outputs when the line foobar is entered on its own line
Let's add line numbers to the file **fragmented** using the nl command

```
$nl < fragmented
```

Notice how it only puts line numbers on lines with entries. Also re-run the command without the <

It should still work. This is because it takes its std input from a file

If no file was specified then it would expect std in from the keyboard. Try it. You will get a flashing cursor. Put some values in followed by return. Notice it numbers each entry. Ctrl-c to end it.

If you wish to number every line of a file then use

```
$nl –ba fragmented
```

NOTE See also cat –b and cat –n and pr -n

If you wish to save this information as it is only being output to the screen then add a > and a filename.

Now let us read std input from the keyboard until a specific marker. We will make the specific marker the word "end"

```
$sort << end
```

Enter some names followed by return

Then the last entry type "**end**"

Notice how it finishes the stdin from the keyboard and outputs the sorted information. If you wanted to store this information

```
$sort << end > sorted
```

If you wanted to line number this output regardless of blank lines

```
$sort << end | nl –ba > sorted_numbered
```

More echoing

You can turn on the backslash escaped character like tab, backspace, formfeed, newline etc, by specifying –e with echo

```
$echo –e "The quick brown fox\tjumps over the lazy dog"
$echo –e "The quick brown fox\njumps over the lazy dog"
$echo –e "The quick brown fox \bjumps over the lazy dog"
```

Task 2 - Octal dumping

Now we will look at the names file in detail. If we use the od command it will output the names in an octal format. This is not very useful.

```
$cd textproc
$od names
```

But if we use the switch –tc or –c (both the same)

```
#od –tc names
```

Notice how the output shows control characters.
\t indicates a tab, \n newline etc.

Check the man pages for more details on the command.

Od can also be used for displaying the contents of a binary file

```
$od –s /bin/ls
```

Now we know how the names file has been formatted we can us other tools to manipulate the spacing

The expand command converts tabs to spaces

We will use this command on the names file

```
$expand names | od –c
```

Notice how all the \t characters have disappeared replaced with default 8 white spaces

The \n newlines are still in there. To change the white space numbers specify the –t1 in the command

```
$expand –t1 names | od –c
```

run that command again but create a new file without the tabs

```
$expand –t1 names > expanded_names
```

Check it with od

```
$od –c expanded_names
```

Now put the tabs back in

```
$unexpand –t2 expanded_names | od -c
```

Now translate the newline /n into white spaces (Carriage returns are /r)

```
$tr '\n' ' ' < names | od –c
```

Now try it without the od command

$tr '\n' ' ' < names

Notice how tr needs the < if it is the first command on the line. Use tr to uppercase the names file

$cat names | tr [a-z] [A-Z]
$cat names | tr "a-z" "A-Z"
$cat names | tr 'a-z' 'A-Z'

Notice how tr can use ' single quote, double quote, or square brackets, it can even do it without brackets

Try the following

$cat names | tr Luke Pete

This will change all entries with Luke to Pete

Deleting data with tr

$tr –d '\n' < names | od -c

Task 3 - Heads and tails, cat and tac

Look at the contents of the number file. It is a sequential numbered up to 30

```
$cd
$cd testproc
```

To look at the last 10 numbers we use

```
$tail numbers
```

To see the last 12 numbers

```
$tail –n12 numbers
```

Or

```
$tail –12 numbers
```

To look at the first ten lines in the file

```
$head numbers
```

If we wanted the middle 10 numbers of a file

```
$head –20 numbers | tail
```

If you want to reverse the file use tac

```
$tac numbers
```

Task 4 - Large documents

Now use the Dracula document and prepare it for printing
$ cd
$ cd textproc

$pr dracula | more

Notice how it now has headers with the date, document name and Page number on it

This document is too wide to print, maybe we need to narrow it down to 60 columns for our new dot matrix printer.

$fmt –w60 dracula | pr

Now make each page have two columns of text each 30 chars wide

$fmt –30 dracula | pr --columns=2 | more

To find out how many lines this document has

$wc –l dracula

If we wish to reduce the size of this document because it has too many lines

$split –l 50 dracula split_dracula_

Notice how you now have some new files called split_dracula_aa etc…

$wc –l split_dracula_aa

This reveals that each file is 50 lines long.

Now create some empty files

$mkdir greptest
$cd greptest
$touch ratas saran cabal jacaw cabin

You should now have those 5 files in the directory greptest

$ls | grep .a [^b]a.

This regexp finds and file that has 5 chars long, where the second and fourth characters are an 'a' The 1st and last characters can be anything, and the 3rd character is not a 'b'

Now **cd ..**

We will look in the dracula text again using grep for entries that begin with the word 'Count'

$cd /home/student/testproc
$grep –w ^Count dracula
Look for all lines that finish with a full stop.

$grep '\.$' dracula

These sort of commands can be useful for removing comments from files, or the reverse.

A comment in a config file usually starts with a # (The pound sign)

To extract only the comment lines

$grep ^# /etc/services

And the inverse is specified with -v

$grep -v ^# /etc/services

Task 5 - Join and Paste

We can join two files together using the join command
$cd /home/student/testproc
$join names surnames

Notice how it has joined the two files together based on matching field 1 the numbers field. Now do the same using a field description in file names use field 3 and use field 2 from file surnames

$join -1 3 -2 2 names surnames

$join -1 3 -2 1 names partialsurnames

Paste joins fields regardless of match

$paste names partialsurnames

Task 6 - Cutting comments

To cut out specific information from a file you can use the cut command. You can specify delimiters and fields within the cut command

```
$cut –d: -f1-3 /etc/passwd
```

This will pull out the first 3 fields in the passwd file

```
$cut –d: -f1,4,5 /etc/passwd
```

This will pull out field 1 4 and 5

To output the mounted filesystems

```
$cut -d ' ' -f1,2 /etc/mtab
```

Extracting fields 1 3 11 and 12 from a uname kernel output. Space as the field delimiter

```
$uname -a | cut -d" " -f1,3,11,12
```

Task 7 - Uniq

Create the following file called **unique** with the following content

$vi unique

```
This line occurs only once.
This line occurs twice.
This line occurs twice.
This line occurs three times.
This line occurs three times.
This line occurs three times.
```

Using the uniq command we can extract unique lines

```
$uniq -c unique
```

```
$sort unique | uniq -c | sort -nr
```

Task 8 - Sorted

We can sort files using the sort command

Sort the names file on the 3rd column which is the surname and let us fold the case.

```
$cd /home/student/textproc
$sort –f –k3 names
```

If you wish to sort with a different delimiter than tab then you must specify a delimiter with the –t option

```
$sort –t: -k1 /etc/passwd
```

Task 9 - Using sed

Copy the following files from /etc to your /root folder

```
$cp /etc/fstab ~
$cp /etc/passwd ~
```

Search for the word **sda** and replace it with **hda** when the line contains the key **swap** in the file **fstab**

```
$cd
$sed ‘ /swap/ s/sda/hda/g’ fstab
```

Search for the word **:** and replace it with **;** when the line contains the key named in file **passwd**

```
$sed ‘/named/ s/:/;/g passwd
```

Delete all commented lines

```
$cd /home/student/textproc
$sed ‘/^#/ d ‘ sedexample
```

Delete line 2 and 3 from the output

```
$sed ‘2,3 d’ sedexample
```

We can use **awk** to extract the **UID** field (Field 3) for only users with the word **root** in their entry. The field delimiter in a passwd is a :

```
$grep –w root /etc/passwd | awk –F : ‘{print $3}’
```

Task 10 - Lab Learning vi

In this section we are going to learn vi using the vimtutor. This lab can be quite long, but vi is an important tool to learn when administering *NIX based machines.

Change directory into your home directory

$cd ~

Now make a directory called vi_tutorial

$**mkdir vi_tutorial**

Change to the directory you have just created

$cd vi_tutorial

Now start the vim tutor

#vimtutor

Now follow the onscreen instructions, all the lab is contained within the file

FIREBRAND

Chapter 2
Lab - Installing software

Use your **Fedora 21** machine for the next part of the lab
Login as student with password root.

Task 1 - Using the du command

Before downloading and installing any packages, let's use the df and du commands to ensure we have enough room for those packages.

Login to the system as the root user. Password is password
From the prompt issue the command:

```
$du -hx /usr/local
```

This will work its way through the entire system and report back the sizes for all files, directories etc.
When it's done, note the space information on the last line here: __________
________________________.
Now issue the following command:

```
$du -h
```

Note the space information on the last line here: _________________________.

Task 2 - Using the df command

While still logged in as root, use the df command to show your disk free amounts by typing the command:

```
$ df -h
```

```
Filesystem     Size   Used   Avail   Use%  Mounted on
/dev/hda5       7.7G  5.1G   2.2G    69%   /
none           125M   0      124M    0%    /dev/shm
/dev/hdb1      75G    54G    17G     75%    /data
/dev/hdd1      75G    53G    18G     74%   /cdroms
/dev/hda7      21G     9.5G  12G      44%   /home/rbrunson/ddrive
#
```

Note: The output shown is from our test system, yours will differ.
To show the file system information along with the file system type, use the following command:

```
$ df -hT
```

```
Filesystem     Type   Size   Used   Avail  Use%  Mounted on
/dev/hda5      ext3   7.7G   5.1G   2.2G   69%   /
none           tmpfs  125M   0      124M   0%    /dev/shm
/dev/hdb1      ext2   75G    54G    17G     75%   /data
/dev/hdd1      ext2   75G    53G    18G    74%   /cdroms
/dev/hda7      vfat   21G    9.5G   12G    44%   /home/rbrunson/ddrive
```

This output shows you as much information about the mounted drives as you can get in a single command.

Task 3 - Installing rpm packages and yum

This lab involves downloading and installing packages. Linux is constantly evolving so many of the packages we use change their versions frequently so it is not practical to list specific packages in this lab.
The instructor will detail which packages are to be used and below are the commands that are to be used with the chosen packages.

On your F21 VM, you will need to become root with the password password

$su –

#wget http://package-name

Check the signature of the file
#rpm –K package-name

To show info on the rpm file
#rpm –qpi package-name

To show/list where the files will be installed
#rpm –qpl package-name

To query what configuration files it will install
#rpm –qpc package-name
Or
#rpm –qp package-name --configfiles

To query the changelog
#rpm –qp package-name –changelog

To install the package
#rpm –ivh package-name

The important switch in the above is the p which states it is querying a package .rpm file

Task 4 - Using the Yum package management tool

Yum works on repositories which are local and online package caches that can be queried and downloaded from.

Carry out the following on the Fedora 21 virtual machine

To see which repos are currently enabled
#yum repolist

Let's search for a package called "**package-name**"
#yum search package-name

To install the package
#yum install package-name

Yum has a main configuration file called **/etc/yum.conf** and a cache directory at **/var/cache/yum**. As yum installs packages into the rpm database, rpm can also be used to query the packages instead of yum.

Query the rpm database for the package we have just installed.

#rpm –qa | grep package-name (the version numbers are just an example, they may be different in your case)

#rpm –qi package-name
#rpm –ql package-name
#rpm –qc package-name
#rpm –q package-name --changelog

You can use rpm tool to query what package installed a certain file

#rpm –qf /etc/fstab
#rpm –qf /etc/hosts
#rpm –qf `which sshd`

To check what files have changed since setup-2.6.10-1.fc8 has been installed
#rpm -V setup-2.6.10-1.fc8

To find out what each value is use
#man rpm

Now remove the package
#rpm –e package-name

Task 5 - Checking library dependencies

Use your **FC21** server for this exercise

Open a command shell and use the ldd command to identify the required shared libraries for the sshd command

#ldd `which sshd` (use the correct quotes! Top left on keyboard)

Shared libraries are managed via a configuration file /etc/ld.so.conf . Look at this configuration file with the more command

#more /etc/ld.so.conf

We can rebuild the **/etc/ld.so.cache** file using the **ldconfig** command
#ldconfig -v

To set the environment variable for developers, first make a directory in your home directory

#mkdir $HOME/libs

Set the variable

#LD_LIBRARY_PATH=$HOME/libs

Check that the variable is set

#set | grep LD

Export the variable

#export LD_LIBRARY_PATH

Check that it is exported

#env | grep LD

To set this permanently edit **/etc/profile** or **~/.bash_profile**

Now unset the variable

#unset LD_LIBRARY_PATH

Check that the variable has been unset

#set | grep LD

Task 6 - Debian Package Management

DPKG

Log into your Debian vm system as root and open a terminal window

Listing packages already installed on the system

#dpkg --list

To find a specific package installed on the system

#dpkg --list | grep passwd

If you wish to know the status of a package

#dpkg --status passwd

To list the files installed by the package

#dpkg --listfiles passwd

To show what package installed a particular file

#dpkg --search /usr/bin/passwd

Task 7 - Using APT

We need to update the package list from the repositories.
The repo file is **/etc/apt/sources.list**. Have a look at the content of this file

#more /etc/apt/sources.list
#apt-get update

Now lets search the package cache for the ssh daemon
#apt-cache search sshd

You can see that it comes back with openssh-server

To install it you would type
#apt-get install openssh-server

To upgrade all the packages currently installed
#apt-get upgrade

To completely upgrade a distribution
#apt-get dist-upgrade

To remove a package
#apt-get remove openssh-server

Lab - Managing processes

Task 1 - Process id and jobs

On your Fedora 21 machine log in as root user
We will start a few processes and manage them through the command line
Open a command shell and change directory to your home.

#cd

Start a process called yes and redirect its out to /dev/null (The bit bucket) the & puts the process in the background

#yes > /dev/null &

Start the top command and put it into the background

#top &

Now let's start an md5sum process to calculate the md5 hash of the first drive on the system.

#md5sum /dev/sda
Notice how this hangs the prompt; it should take a long time to complete this task.

Let's stop the process and background it

To stop the process

CTRL+Z

This stops the process but we must restart the job in the background.

To see the jobs numbers

#jobs
As this was the last job that was stopped, we can just use the bg command with out any options

#bg

NOTE If the job was not the last stopped, you can use the jobs command to show a list of job numbers, and then use **#bg JOBID**

Now list the current jobs running and stopped

#jobs
[1] running yes > /dev/null &
[2] + stopped top
[3] - running md5sum /dev/sda &

We will bring the top job to the foreground

#fg 2

This should now display the top tool on screen

CTRL+Z will stop the job and then background the job

#bg

To list the process IDs of the current processes running in the current shell

#ps

Identify the process ID for the yes process, in this example mine is 27522
To kill this process with a SIGTERM (-15)

#kill 27522

If that failed, you can use a SIGKILL (-9)

#kill -9 27522

We will now kill all process by process name using a user defined signal USR1
Make sure the apache process is running

#service httpd start
# ps –fe | grep http
#killall -USR1 httpd
#ps –fe | grep http

To list all process running on the system, issue the following command

#ps -ef

To find the process ID of a specific process named

#ps -ef | grep bash

Another useful command is the **pstree** command which shows a tree structure of the cascading process IDs.

#pstree -p
Open another command shell run the **top** command
#top

This opens up a tool that shows the top processes running on your system. This tool can be used to kill processes, renice processes, sort and various other process management. Press the **h** command to get a list of help

Task 2 - Nice and Renice

Switch back to the original shell. Identify the process ID of the md5sum command running.

```
#ps -ef | grep md5sum
```

if this md5sum is not running, lets start it again

```
#md5sum /dev/sda
#ps –fe | grep md5sum
```

We shall change the process priority

Its current Nice value should be 0
Check this with the following command

```
#ps axo pid,ni,comm | grep md5sum
```

My process ID is 27528

```
#renice -5 27528
```

At the risk of grinding the system to a halt do not go too mad on assigning negative numbers, max value is -20. Only root can assign a negative number.

```
#renice -10 27528
```

To make the process nicer to its surrounding processes

```
#renice +10 27528
```

You can start a process with a different nice value. The default is +10

```
#nice md5sum /dev/sda
```

To go negative you must put a double negative in front of the nice value

```
#nice --10 md5sum /dev/sda
```

Chapter 3

Task 1 - Investigating Hardware

On your Fedora 15 Machine, login as root and open up a command shell and change to the **/proc** directory

#cd /proc
#ls

In this directory you can see the pseudo file system that contains such information as IRQ, cpu information, DMA channels etc. Let's have a look at some of these:

#more cpuinfo
#more interrupts
#more dma
#more ioports
#more modules
(these commands can also be run on the host FC15 machine)
or

#lsmod
#more swaps

or

#swapinfo –s
#more uptime

or

#uptime

The file **cmdline** in **/proc** directory holds details of the arguments passed by GRUB or LILO to the booting kernel.

#more cmdline

In the **/proc** folder there are a set of numbered directories, now change into directory 1

#cd 1
#ls –l

In this folder you can see the attributes of the process ID PID 1 the init process. Notice how **exe** file links -> to **/sbin/init**. This is where tools like **top** and **ps** get their output information from.

A very useful file in here is **cmdline** which contains the arguments that where passed to the init process when it starts

# more cmdline

To see what the other files in this directory are try using the man page

#man 5 proc

USB
Detailed information about the USB subsystem can be extracted from these directories as well, see **/proc/bus/usb**

#cd /proc/bus/usb
#more devices

The devices file holds information about the usb hubs and devices attached, a tool that extracts the same information is the **lsusb** tool.

#lsusb –v

As can be seen it contains the same information as the devices file.

PCI Bus
The PCI bus can be enumerated by looking at the /proc/bus/pci directory structure

#more /proc/bus/pci/devices

And similarly the tool is **lspci**

#lspci –v

For a bus centric view of the bus including IRQ numbers use the switch –b

#lspci –b
And for a tree centric view use –t

#lspci –t

Notice how the MAC address of the network card is not shown

DMESG
The **/var/log/dmesg** file contains the output from the Kernel ring buffer (messages output at boot time before the sys logging service starts). You can either cat this file or use the dmesg tool to output it

#more /var/log/dmesg
#cat /var/log/dmesg | grep -i cdrom

or

#dmesg

Lab - Managing File Systems

Task 1 - Partition the Disk with fdisk

Use the **Fedora 21 virtual machine** for this Lab.

Log in as student and change user to root.

Have a look at the current setup of your disk

#fdisk –l /dev/sda

We will see we have the following partitions set

/dev/sda1 /boot
/dev/sda2 /
/dev/sda3 /home

Now create another partition. We may want to create a few more partitions after this one so we need to create an extended partition as we only have room for one more primary partition

In this lab we will create the following setup

/dev/sda1	/boot	200MB
/dev/sda2	/	5GB
/dev/sda3	/home	1GB
/dev/sda4	Extended partition	From end of sda3 to end of disk
/dev/sda5	/north	1GB

First create the extended partition. We'll use fdisk to accomplish this

List out your current disk format
#fdisk –l

To check the sizes in a more readable format
#parted -l

Begin the partitioning of your disk with:
# fdisk /dev/sda

1. Press the "p" key & Enter to show the disk's partition table
2. Press the "n" key & Enter to create a new partition
3. Press the "e" key & Enter to make it an extended partition
4. It will prompt you for the partition number, Press "4" and Enter
5. You'll be prompted for the beginning cylinder, take the default by hitting Enter
6. Accept the ending cylinder by hitting Enter
7. Press the "n" key & Enter to create a new partition
8. When prompted for the starting cylinder, take the default by hitting Enter
9. When prompted for the size or ending cylinder, enter "+1G" and hit Enter
10. Press the "p" key & Enter to display the disk's partition table
11. Press the "w" key to commit your disk configuration

If you get an error stating Re-reading the partition table failed…..
The kernel still uses the old table

To check the kernel partition information

```
#more /proc/partitions
```

Notice the new sd4 and sd5 partitions are not there. Run the following command

```
#partprobe
#more /proc/partitions
```

All should be fine now

Task 2 - Make a File System on the Disk

Now let's make a file system on the new partition.

Login as root
Make a file system with the command:

```
#mkfs.ext4 /dev/sda5 -
```

Make a mount point with:

```
#mkdir /north
```

Mount the new partition to see if it's ok with:

```
#mount -t ext4 /dev/sda5 /north
```

Verify the size of the new partition with:

```
#df -h
```

Change the mounting options with the command:

```
#tune2fs -c 0 /dev/sda5
```

Create a sample file on your new file system with:

```
#touch /north/file1
```

```
#ls -l /north/file1
```

Unmount the file system with:

```
#umount /north
```

Task 3 - Setup the File System for User Access

Now let's edit the /etc/fstab file to mount this disk on user request, and set the options so that any user can mount or umount the file system.

Edit the /etc/fstab file with:

```
#vi /etc/fstab
```

Navigate to the last line in the file

Open a new line below the current line with the "o" key

Add the following line to the file, using the TAB key for spacing:

```
/dev/sda5 /north ext4 defaults,users,noauto 0 0
```

Exit and save the file with **esc:wq!**

Test the mount the new entry with

```
#mount /north
#df -h
```

Unmount the new entry with:

```
#umount /north
#df -h
```

Change over to your normal user with:

```
#exit
```

Mount the new entry with:

```
$mount /mnt/sda5
$ df -h

$umount /mnt/sda5
$df -h
```

As you can see, the "users" option allows anyone to mount and umount the entry in the fstab file, and upon your next reboot, you'll see the file system is not automatically mounted, you'll have to mount it manually, due to the "noauto" option in the fstab. Removing the "noauto" keyword would allow the file system to automatically mounted on system boot.

Task 4 – Creating a Swap file

Use the Fedora 21 virtual machine for this Lab. You will need to be root.

We currently have no swap

To see this
```
#free -mt
#more /proc/swaps
```

We will now create a swap file called extraswap stored on the root of the filesesyem which resides on the filesystem and add it to the swap system
First create an empty file called extraswap with a blocksize of 1024 bytes by 1024 blocks long . To do this use the dd command as shown below

```
#dd if=/dev/zero of=/extraswap bs=1024 count=1024
```

Now convert this empty file to a swap file with the mkswap command

```
#mkswap /extraswap
```

Now you must enable the swap file with the swapon command

```
#swapon /extraswap
```

Monitor swap space with the free command

```
#free –mt
```

To make this available after a reboot

```
#vi /etc/fstab
/extraswap  swap  swap  defaults 0 0
```

Do not reboot just in case you have made an error in this file.
Test it while you are logged in. First lets remove the swap file we added on the fly

```
#swapoff /extraswap
#free –mt
```

Now lets add/test our swap entry in /etc/fstab

```
#swapon –a
#swapon –s
#free -mt
```

Chapter 4
Lab – Managing Files

Task 1 - Creating User Quotas

On your f21 vm login as root.

You should have the following setup now

/dev/sda1	/boot	200MB
/dev/sda2	/	5GB
/dev/sda3	/home	1GB
/dev/sda4	Extended partition	From end of sda3 to end of disk
/dev/sda5	/north	1GB
/		

We will create our quotas on the /dev/sda5 partition.

Create the two quota database files on the root of this partition

#touch aquota.user ; touch aquota.group

Change the permission to allow modification of the quotas for root users

#chmod 660 aquota.*

Edit the /etc/fstab to change the /dev/sda5 entry to map it to /home

#vi /etc/fstab
<UUID of /dev/sda5> /home defaults,usrquota,grpquota 0 0

Check that quotas are now set by using the mount command to view the options set

#mount

Create a new user

#useradd richard
#passwd richard

Enter a new password

Populate the quota files aquota.user and aquota.group by running the following command. Notice how the file size has changed

#quotacheck –avug

#ls –l

Edit the users quotas

#edquota –u richard
This will open up a vi session with the framework to set the user's hard and soft limits

Should look something like below

```
Disk quota for user richard uid(1001)
Filesystem     blocks  soft     hard    inodes          soft      hard
/dev/sda5      0        0       0       0               0         0
```

Change it to

```
Disk quota for user richard uid(1001)
Filesystem     blocks soft      hard    inodes          soft     hard
/dev/sda5      0        0       0        0              10       12
```

This will set the number of files the user can create to 10 to report and 12 to enforce

Turn the quotas on

#quotaon -a

Let's change to the user richard

# su – richard

Enter the password and create new files using the touch command
#touch file1
#touch file2
#touch file3
Etc….

Until you reach the soft limit, it will then report soft limit reached
Carry on creating until you reach the hard limit

Type exit to return to the root shell
#exit

Lets check the quotas

#quota richard

#repquota –a

Setting Grace periods
Grace periods allow you to enforce quotas after a given time period

#edquota –t
This opens a vi session as before and here you can enter a days , hours minutes or seconds entry per filesystem.

Task 2 - Setting Permissions

Use your Fedora 21 machine and log in as root.
Open a command shell and change directory to /home

```
#cd /home
```

Create a shared directory

```
#mkdir shared
```

Now do a long listing to view the contents of the /home directory

```
#ls -l
```

The default permissions should be as follows

```
drwxr-xr-x root root shared
```

Create two new users called **david** and **paul**

```
#useradd -G users david
#useradd paul
```

David is a member of the **users** group and paul is not

Set a password

```
#passwd david
#passwd paul
```

Now change the group ownership of the shared directory

```
#chgrp users /home/shared
```

Now stop any user not in the defined group users from accessing this directory

```
#chmod 750 /home/shared
```

Now test this theory

Change user to david

```
#su - david
```

Check your group membership

```
$id
```

Now try to change directory to /home/shared

$cd /home/shared

You should successfully change directory to the shared directory

Now try to create a file in the directory

$touch davids_file

To exit back to root

$exit

You do not have permissions to create anything in this directory as the directory permissions are rwxr-x---

Change users to paul

#su - paul

Check your group membership

$id

Now try to change directory to /home/shared

$cd /home/shared

To exit back to root

$exit

Task 3 - Setting SGID on a directory

As root user, working in the **/home** directory

Now change the shared directory so the users group can write in it.

```
#chmod 770 /home/shared
```

Now change user to david

```
#su - david
```

Change directory to the shared directory

```
$cd /home/shared
```

Create a new file called davids_file

```
$touch davids_file
```

Do a long listing on the directory

```
$ls -l
```

Notice how the file you have just created has the owner and primary group of the user david. This causes a problem as only david may have access to the file.

```
-rw-rw-r-- david david davids_file
```

Now let's set an SGID on the directory

Exit back to the root user

```
$exit
```

Set the SGID on the directory

```
#chmod 2770 /home/shared
```

Now when you do a long directory listing

```
#ls -l /home
drwxrws--- root users shared
```

Note the **rws** on the group

Switch user to the user david

```
#su - david
```

Change to the shared directory

```
$cd /home/shared
```

Create a file called newfile

```
$touch newfile
```

Look at the permissions of the newfile with a long list

```
$ls -l
```

Notice now how the file newfile has the group ownership of the directory

```
rw-rw-r-- david users newfile
```

Task 4 - chown and chgrp

Remaining in the shared directory we shall change the ownership and group ownership of the file.

To change the ownership of the file and group in one go

#chown david.david newfile

To change just the group

#chown .users newfile
or
#chgrp users newfile

Task 5 - umask

The umask sets your default create permissions when creating files and directories.

Make sure you are in your home directory

#cd

To view the current umask setting

#umask

The output should be **0022**. This gives an effective file creation of **rw-r--r--**

And a directory creation **drwxr-xr-x**

Test this by creating a directory

#mkdir umask_test

Now look at the permissions

#ls -ld umask_test

Now create a file called umask_file and look at the permissions

#touch umask_file

#ls -l umask_file
The *umask* value is subtracted from these default permissions after the function has created the new file or directory. Thus, a directory will have permissions of *775* by default, a file *664*

Set the umask to 002

#umask 022

Now create a directory and a file

```
#touch newfile1
```

```
#mkdir newdir1
```

Now view their permissions

```
#ls –l
```

Chapter 5 Booting

Task 1 – Changing boot target and rescue mode

On your machine running fedora 21 system, switch to the multi-user target manually without rebooting.

```
#systemctl get-default

#systemctl isolate multi-user.target
```

Log into a text-based console as root and configure your system to automatically boot into the multi-user target after a reboot, then reboot your system to verify.

```
# systemctl set-default multi-user.target
# systemctl reboot
```

You should now have the text based console login

To boot your system into rescue mode
Reboot your system, then from within the boot loader menu, boot into the rescue target using the following instructions.

```
# systemctl reboot
```

Interrupt the boot loader when the menu appears by pressing any key.
Move the selection to the default entry (the first one) using the cursor keys.
Press e to edit the current entry.
Move the cursor to the line that starts with linux16.
Move the cursor to the end of the line (line may wrap round), and append the following text:

```
systemd.unit=rescue.target
```

Press Ctrl+x to boot using the modified configuration.

When prompted for the root password, enter password
You are now in rescue mode.

Set the default systemd target back to the graphical target**.**

```
# systemctl set-default graphical.target
```

Press Ctrl+d to continue booting into the (new) default target.

Chapter 6 - Lab User interfaces and desktops

Task 1 - Localisation and Internationalisation

To view the current locale setting use

```
#locale
```

Timezone information is stored in /etc/localtime and not in the locale variables. This file contains binary data of the time zone fetched from /usr/share/zoneinfo

```
#md5sum /etc/localtime
```

```
#md5sum /usr/share/zoneinfo/GB
```

The output from these should be the same showing it is the same file

The /etc/sysconfig/clock file on Fedora contains timezone information as well

```
#cat /etc/sysconfig/clock
```

You can configure the timezone through the command line using

```
#tzselect
```

Follow the prompts

For a GUI version on Fedora you can use

```
#system-config-date
```

The locale setting is stored in /etc/sysconfig/i18n and /etc/sysconfig/keyboard

```
#cat /etc/sysconfig/i18n
```

Chapter 7 - Lab Admin Tasks

Task 1 - User and group account management

We shall first create a couple of accounts using the useradd command using the defaults that are set in **/etc/login.defs** and **/etc/default/useradd** files

To view the current defaults

#useradd -D

This is the same as

#cat /etc/default/useradd

Create an account for peter

#useradd peter

To view the information about the account you have just created

#id peter

Now set a password for peter

#passwd peter

Look at the home directory for peter

#cd ~peter

The contents of the directory should have the contents of the /etc/skel directory copied to it. The files .bashrc .bash_profile and .bash_logout as a minimum.

#ls -la

Now add a user but change the default values. A comment field of Mike Smith, a primary group of users and a secondary group of wheel . He will also have a default shell of /bin/ksh

#useradd -c “Mike Smith” -g users -G wheel -s /bin/ksh mike

Check this with the following commands

#grep wheel /etc/group

#grep mike /etc/passwd

We need to lock the peter account

#passwd -l peter

To unlock the account

#passwd -u peter

To view the current passwd settings for the user peter

#passwd -S peter

Another way of disabling peters account would be to change his shell to /bin/false

#vi /etc/passwd

Locate peters line in the file and change **/bin/bash** to **/bin/false**

Save and exit with **escape Shift ZZ**

Hit the **Ctrl+Alt+f3** key to switch to a console

Now try and log in with peters username and password

Switch back to X environment with **Ctrl+Alt+f1**

To set the max number days for the password for peter.

#passwd -x 6 peter or #chage -M 6 peter

To view the settings

#passwd -S peter

You can also **#cat /etc/shadow** as this where the changes are made.

A better output can be seen with chage

#chage -l peter

Now modify peters primary group and change it to adm

```
#usermod -g adm peter
```

```
#id peter
```

Another option to see peters groups is

```
#groups peter
```

We need to create a group called finance with a gid of 555

```
#groupadd -g 555 finance
```

To delete the group we have just created

```
#groupdel finance
```

And finally to delete the user peter and his home directory

```
#userdel -r peter
```

To perform a sanity check on the passwd and shadow files

```
#pwck
```

Finally, to modify the defaults in the /etc/default/useradd file without using vi

```
#useradd -D -s /bin/csh
```

View the change with

#useradd -D

Note the useradd and usermod commands use the same switches

-s shell

-c comment

-g primary group

-G secondary group

Permissions on the **/etc/passwd** and **/etc/shadow** files

It is important to know these settings

#ls -l /etc/passwd

Document your findings here ___________________

#ls -l /etc/shadow

Document your findings here____________________

Working with the skeleton directory

We shall make some changes to the **/etc/skel** directory

#cd /etc/skel

```
#ls -la
```

Make a couple of directories in this directory

```
#mkdir lib
#mkdir bin
```

Edit the **.bash_profile**

```
#vi .bash_profile
```

Add the line underneath the PATH

```
LD_LIBRARY_PATH=$HOME/lib
```

Save and Exit with Escape Shift +ZZ

Now create a new user megan

```
#useradd megan
```

Change directories to her new home directory

```
#cd ~megan
```

Check that the changes you have made to the skel directory are invoked into the new users directory

```
#ls -l
```

```
#more .bash_profile
```

We shall now make a similar change to the **/etc/login.defs** file

```
#vi /etc/login.defs
```

Scroll down through this file until you find **PASS_MAX_DAYS**

Change this value to **60**

```
#useradd arnold
```

Now view the changes you have made using the **chage** command. The password max number of days should now be 60

```
#chage -l arnold
```

Task 2 - Scheduling jobs

First we shall use the at scheduler

```
#at now + 4 minutes
at>eject
ctrl+d to submit
```

To view the job use

```
#atq
```

The job queue exists in the directory /var/spool/at

```
#cd /var/spool/at
```

Now cat the a???????? file that exists in this directory. This is the spooled job file

To remove the job.

The job number is ascertained by the atq command

```
#atrm 1
```

To schedule a cron job

```
#crontab -e
```

This will open a vi session. We shall add a couple of entries to eject and close

The tray on the cd

```
#Min    Hour   DoM    Mon    DoW    Command
0-58/2  *      *      *      *      eject
1-59/2  *      *      *      *      eject -t
```

Save the settings with **:wq!**

To view the current crontab

#crontab -l

The crontab is installed in the **/var/spool/cron**

To remove the cronjob, either edit the crontab with crontab -e and deleted the entries not required or

#crontab -r

You can edit another user's crontab as root user

#crontab -u mike -e

And similarly to delete the users crontab

#crontab -u mike -r

The system crontab that runs the cron.hourly, cron.daily etc is in the /etc directory

```
#more /etc/crontab
```

In this file it list the directory entries for

/etc/cron.hourly/

/etc/cron.daily/

/etc/cron.weekly/

/etc/cron.monthly/

Have a look at the scripts in those directories

```
#cd /etc/cron.daily
```

```
#more logrotate
```

Task 3 - Maintain system time

To view the current OS time enter

```
#date
```

To view the current BIOS time

```
#hwclock -r
```

The OS clock can be set to the current BIOS clock

```
#hwclock --hctosys
```

If the bios clock has a different timezone use the localtime option

```
#hwclock --hctosys --localtime
```

To copy the OS clock back to the BIOS

```
#hwclock --systohc
```

This is actually done in the shutdown scripts

NTP

First check if NTP is running

```
#service ntpd status
```

You could also use

```
#ps -ef | grep ntp
```

If it is started, stop it using the following

```
#service ntpd stop
```

Check the configuration file for the ntp servers used

```
#vi /etc/ntp.conf
```

Scroll down to the lines and check content

```
server 0.fedora.pool.ntp.org
server 1.fedora.pool.ntp.org
server 2.fedora.pool.ntp.org
```

Now start the service to maintain the time

```
#service ntpd start
```

To view the current offset of the system clock

#ntpdc -c loopinfo

To view the remaining increments to sync the time

#ntpdc -c kerninfo

Make sure the service is enabled for runlevels 345

#chkconfig --list | grep ntp

If not then

#chkconfig --levels 345 ntpd on

The ntptrace command can be used to show the route your time has taken

#ntptrace 0.fedora.pool.ntp.org

Task 4 - System Logging

The main configuration file is **/etc/rsyslog.conf** (/etc/syslog.conf on older machines)

Edit this file

#vi /etc/rsyslog.conf

Find an empty line and put the following entries in

```
*.*                      /dev/tty5
local1.!=info            /var/log/locallog
```

Save and exit the configuration file with **Shift ZZ**

Restart the syslog daemon

#service rsyslog restart

Switch to the tty5 console with **CTRL+ALT+F5**

You should see your console output being displayed

Switch back to your X environment with **ALT+F1**

Now send an entry to the **/var/log/locallog** file

```
#logger -p local1.crit “Entry from logger command”
```

Check that it has put an entry in /var/log/locallog

```
#tail /var/log/locallog
```

Chapter 8 - Lab - Network Fundamentals

In the following exercises the target network is 172.16.0.0 and the IP address used is 172.16.0.5. Check with the instructor which network and address is to be used for these exercises.

Task 1 - Fundamentals of internet protocols

Use the **/etc/services** to find out the following port numbers

grep -w 21 /etc/services

Now try ports 80, 53, 110, 143, 25, 69

Network monitoring

We shall use Wireshark to view the network traffic

First we need to install it

#yum install wireshark

Select y to download and install it

Once installed start it with

#wireshark &

Firstly, identify your Ethernet interface. This can be done by running the **ifconfig** command. The interface should start with the letter e but may be em? or en?

Start a capture on your Ethernet interface and then do the following scans. Collect and visualize the packets

Use nmap to scan your localhost for what ports are open on TCP

Firstly with a TCP Connect Scan

#nmap -sT localhost

Now with a Syn Scan

#nmap -sS localhost

Now try scanning the server at 172.16.0.5 with the same scans

Alternatively use whatever addresses are given by the instructor

#nmap -sT 172.16.0.5 (these may differ)

#nmap -sS 172.16.05

Check your current route on your machine

#netstat -rn

Now try with the route command

#route

To delete the default route

#route del default gw 172.16.0.1

To re add the route

#route add default gw 172.16.0.1

The arp cache can be viewed with

#arp -a

Look for the IP address 172.16.0.1 in the arp cache output.

Using the MAC address listed next to 172.16.0.1 we shall set this as a static arp entry

#arp -s 172.16.0.1 XX:XX:XX:XX:XX:XX

Where XX:XX:XX:XX:XX is the mac address of the default gateway

Now let's check connectivity

#ping 172.16.0.1

If that fails then you probably have the wrong mac address entered

Delete the static entry with

#arp -d 172.16.0.1

Task 2 - Basic Network Configuration

Login as root onto your F21 VM

Your current network settings will be set with DHCP . This is set in the

/etc/sysconfig/network-scripts/ifcfg-<Interface> on a Fedora box.

```
# ifconfig -a
```

```
#cat /etc/sysconfig/network-scripts/ifcfg-<interface>
```

To stop the network daemon

```
#service network stop
```

Now let's set the IP address manually to 172.16.0.X where X is your assigned IP

```
#ifconfig <Interface> 172.16.0.X netmask 255.255.255.0 up
```

Check your settings

```
#ifconfig -a
```

To down the network card

```
#ifdown <Interface>
```

Now restart the network

```
#service network start
```

The ip address can also be changed using the new ip command

```
#ip addr show
```

```
#ip addr add 172.23.0.X/24
```

```
#ip addr show
```

Task 3 - Basic Network Troubleshooting

On your physical F15 system login as root.

Use ping to check connectivity to 172.16.0.5

```
#ping 172.16.0.5
```

Hit **ctrl + C** to cancel

Now send a larger data payload and only 3 packets

```
#ping -s 1024 172.16.0.5 -c 3
```

To see all hops to a specific host use traceroute

```
#traceroute www.ocf.co.uk
```

To turn off names resolution and speed up the test

```
#traceroute -n www.ocf.co.uk
```

To check dns resolution you should use dig and host

For the ocf.co.uk domain let's find out the following

Mail exchanger

```
#dig ocf.co.uk MX
```

Name Servers

```
#dig ocf.co.uk NS
```

Host www.ocf.co.uk

```
#dig www.ocf.co.uk
```

To resolve against a different nameserver i.e. 172.16.0.5

```
#dig mail.ocf.co.uk @172.16.0.5
```

Using the host command

```
#host -t MX google.com 172.16.0.5
```

Task 4 - Client Side DNS

Names resolution servers IP addresses are stored in /etc/resolv.conf

#cat /etc/resolv.conf

Let's append the 172.16.0.5 address to the end of the /etc/resolv.conf

#echo "nameserver 172.16.0.5" >> /etc/resolv.conf

#cat /etc/resolv.conf

Now edit the /etc/hosts

#vi /etc/hosts

Add a line as follows after the last entry line

172.16.0.5 server.lpiclass.lab server

172.16.0.5 www.bbc.co.uk bbc

Save and exit with :wq!

Now let's see if it resolves

#ping server

#ping www.bbc.co.uk

These should give back the IP address of 172.16.0.5

The order of resolution is done via a file called /etc/nsswitch.conf

#cat /etc/nsswitch.conf

In this file there is a line as follows

hosts: files dns

This means that /etc/hosts is checked first then dns via the /etc/resolv.conf

Task 5 - The MTA

Editing the aliases

#vi /etc/aliases

At the end of the list is a commented out line

#root: marc

Uncomment it and change the name to peter

root: peter

Save and exit vi with **:wq**

Re read the aliases file with

#sendmail -bi

or

#newaliases

Both do the same thing

Now create a mail to root from luke (this assumes you have a user called luke, if not create him with

#useradd luke

#passwd luke

Enter a password twice

#su - luke

$mail root

Enter a subject in the subject line

Subject : Hello root

This message is a message to root forwarded to peter

CTRL+D

Press CTRL+D to submit message

Type exit to return to the root user

Now change user to peter

#su - peter

$mail

This should list a set of numbered mail

To read a mail message type the number followed by return

&1

Now type q to quit

Type exit to return to the root shell

$exit

Chapter 9 Lab - Shell Scripting and Data Management

Task 1 - Setting your shell environment

To view your current shell settings (not exported)

```
#set
```

To view all shell environment variables (exported)

```
#env
```

Let's set the some environment variables

For a user the environment variables are defined in **~/.bash_profile**

For the system wide environment variables look in **/etc/profile**

```
#cat /etc/profile
#cat ~/.bash_profile
```

We are going to define the LD_LIBRARY_PATH variable for a user who is a

developer. He would most likely have a directory in his home directory called libs and let's create a directory for his binaries called mybins.

Mike is the developer (providing you left his account on in the user administration lab)

```
#su - mike
```

Type in the password for mike

```
$cd ~
$mkdir libs
$mkdir mybins
```

Now edit the .bash_profile

```
$vi .bash_profile
```

In this file you should see a line PATH=$PATH:$HOME/bin

Modify the PATH to read

```
PATH=$PATH:$HOME/bin:$HOME/mybins
```

Underneath the PATH add the following line

```
LD_LIBRARY_PATH=$HOME/libs
```

Save and exit.

At the moment this change hasn't been made

```
$echo $LD_LIBRARY_PATH
```

It should be set to null

Now we need to re-read the .bash_profile. This can be done in many ways. The first is to logout and log back in. Or use either source or . to re-read the file

$source .bash_profile or **$. .bash_profile**

$echo $LD_LIBRARY_PATH

$echo $PATH

Logout of mikes account

$exit

The exit variable

The exit variable is used to show how the process exited. 0 for success and 1 for failure

Let's use the test command to see if a file exists

#test -f /etc/passwd

#echo $?

The output should be 0

Now check for a file you know doesn't exist

#test -f /etc/testfile

#echo $?

The output should be 1

Other Special variables include $$ which shows the current shells process ID

#ps

This will show you your bash shells process ID

#echo $$

This will do the same but only give you the process ID of the shell

The sequence command

The seq command can be used to output a sequence of numbers

#seq 10

Will output a list from 1-10 inclusive

If you wish to select a start point i.e. from 10 through 40

#seq 10 40

To pad the leading zeros

#seq -w 90 110

Task 2 - Creating functions

Create a simple function

#fun () { echo "This is a function"; echo; }

To view all the functions defined in the shell

#declare -F

To view the contents of one of the functions

#declare -f fun

To remove the function

#unset fun

Now lets create a script with a 2 functions inside it

#vi funky

Inside vi type the following

#!/bin/bash

JUST_A_SECOND=1

funky ()

{ # This is about as simple as functions get.

```
echo "This is a funky function."

echo "Now exiting funky function."

} # Function declaration must precede call.

fun ()

{ # A somewhat more complex function.

i=0

REPEATS=30

echo

echo "And now the fun really begins."

echo

sleep $JUST_A_SECOND # Hey, wait a second!

while [ $i -lt $REPEATS ]

do

echo "----------FUNCTIONS---------->"

echo "<------------ARE-------------"

echo "<------------FUN------------>"

echo

let "i+=1"

done

}


# Now, call the functions.


funky

fun

exit 0
```

We need to make this script executable

```
#chmod 500 funky
```

Run the script

```
#./funky
```

Notice how the functions are only available within the shell

```
#declare -F
```

Change the permissions of the funky script back to read only

```
#chmod 400 funky
```

When we try to run this script as previous

```
#./funky
```

You should get a permission denied statement

If the script is not executable, you can call the script with the shell to run it in and it should execute it

```
#sh funky
```

Task 3 - Simple Scripts

On the command line type the following

#read a b c

At the blank prompt type in the following with a space between them

1 2 3 4 5 6

Now let us see what is in each variable

#echo $a

____

#echo $b

____

#echo $c

____

Now echo them all out

#echo $c $b $a

____

We are now going to create a script based on the above syntax

#cd ~

#vi numbers

```
echo ‘ 1 2 3 4 5 6 ‘ | while read a b c; do
echo Result $c $b $a;
done
```

Save and exit the script **Esc Shift ZZ**

Now run the scripts

#sh numbers

What is the output from this script. Can you explain the results?

____

Variables and Exporting

Let’s investigate the behaviour of variables in shells

Set the following variable

#NAME=Richard

#echo $NAME

Now create a new shell

#bash

Echo out the NAME Variable again

#echo $NAME

Exit the shell

#exit

Now export the Variable

#export NAME

See if the variable is available in the subshell

#bash

#echo $NAME

#exit

Create the following script

#vi variable_test

```
#!/bin/bash
echo "The current \$NAME is set to $NAME"
NAME=Peter
echo "Name currently set in \$NAME is $NAME"
```

Save and exit vi **:wq!**

Run the command

#sh variable_test

What do you expect the variable to be now? Richard or Peter

#echo $NAME

It should be **Richard** as the value stored in the script is lost when the shell exits.

Task 4 - Simple SQL Data Management

We are going to use mysql for this lab

Check that your mysql service is running

```
#service mysqld status
```

If it is not started start it

```
#service mysqld start
```

Log into your mysql server

```
#mysql -u root
```

To show the current databases on the system

```
mysql>show databases;
```

In the output you should see a database called information_schema. Change databases to this database

```
mysql>use information_schema;
```

To list the tables

```
mysql> show tables;
```

This will output the tables for the information_schema database

To view the construct of a specific table

mysql>describe triggers;

Swap database to the mysql database

mysql>use mysql;

List the tables in this database

mysql>show tables;

mysql>describe user;

The users for the database are stored in user table

mysql>select host,user,password from user;

Now we will create our own database

mysql>create database books_shop;

Now create a table inside this database called **dummy** and then delete it

```
mysql>use books_shop;

mysql>show tables;

mysql>create table dummy ( first_name CHAR(20) , last_name CHAR(20) );

mysql> show tables;

mysql>describe dummy;

mysql>drop table dummy;

mysql>show tables;

mysql>exit;
```

Now create a sql script file with vi from the shell command prompt

```
#vi sql_book_script
```

```
CREATE TABLE books

(

title CHAR(50) NOT NULL,

author CHAR(30) NOT NULL,

publisher CHAR(30),

topic CHAR(20),

comment CHAR(100),

price FLOAT

)
```

Save and exit this with **Shift ZZ**

To execute this script (chmod 755)

```
#mysql books_shop -u root < sql_book_script
```

```
#mysql -u root

mysql>use books_shop;

mysql>show tables;

mysql>describe books;
```

We shall add a book to the table

```
mysql>insert into books
->values ("Hacking for life", "Luke Crowe", "Crowes Press",
->"Biography", "What can I say?",99.95);
mysql> select * from books;
```

Now delete that entry

```
mysql>delete from books where title="Hacking for life";
mysql>select * from books;
mysql>exit;
```

Now we shall populate the books table from a script

The script is available in the /home/student/lpidocs/mysql folder or you can create it below

```
#vi populate_books.sql
INSERT INTO books
VALUES ("MySQL", "Paul DuBois", "New Riders", "MySQL",
"Excellent book, but doesn't cover Java API", 49.99);
INSERT INTO books
VALUES ("Beginning XML", "David Hunter", "Wrox", "XML",
"Well recommended, fairly comprehensive", 39.99);
INSERT INTO books
VALUES ("Java How to Program", "Paul Deitel", "Prentice Hall", "Java",
"Good textbook, extremely detailed", 68.00);
INSERT INTO books
VALUES ("Thinking in Java", "Bruce Eckel", "Prentice Hall", "Java",
"Well written, free on the web", 0.00);
INSERT INTO books
VALUES ("The Java Programming Language", "Ken Arnold", "Addison Wesley",
"Java",
"Considered to be from the source", 37.95);
INSERT INTO books
VALUES ("Learning Perl", "Randal Schwartz", "O'Reilly", "Perl",
"Not a bad start", 29.95);
INSERT INTO books
VALUES ("Programming Perl", "Larry Wall", "O'Reilly", "Perl",
"Usually considered THE reference", 44.95);
INSERT INTO books
VALUES ("Effective Perl Programming", "Joseph Hall", "Addison Wesley", "Perl",
"Great tips, not for beginners", 34.95);
```

Save and exit with **:wq!**

Now populate the books by running the script

```
#mysql books_shop -u root < populate_books.sql
```

Log back into the database

```
#mysql books_shop -u root
```

View the table

```
mysql>select * from books;
```

Show the output ordered by price in ascending order

```
mysql>select * from books order by price asc;
```

And descending

```
mysql>select * from books order by price desc;
```

Now exit

```
mysql>exit
```

Chapter 10 - Lab - Security

Task 1 - Performing Security Admin Tasks

Finding all SUID programs

```
#find / -perm +4000
```

To find all SGID programs

```
#find / -perm +2000
```

Or if you want to find them all

```
#find / -perm +6000
```

Finding files that are hidden with a dot prefix

```
#find / -name ".*"
```

Password ageing

Password ageing defaults are set in /etc/login.defs file, but can be overridden at the command line

To view the current password ageing information of a user

```
#chage -l root
```

or

```
#passwd -S root
```

To change the maximum password

```
#chage -M 60 peter
```

Check the change

```
#chage -l peter
```

A sanity check can be performed on the /etc/passwd and /etc/shadow files

```
#pwck
```

Running commands as another user

The su command allows you to execute or elevate your privileges to that of another user

Log in as a peter on tty2 (ctrl+alt+f2)

```
$su - -c ifconfig
```

Now type in the password for root

Let’s look at the sudo command. This allows you to select which users can execute what commands. We are going to let peter execute networking commands

Switch back to the X window environment (ctrl+alt+f1) where you are logged in as root

#vi /etc/sudoers

Scroll down to the line **User_Alias**. Remove the comment **#** from in front of the line.

Remove the users that are currently listed. Now add a normal users name to the list.

It should look like

User_Alias ADMINS = peter

Scroll down to **Cmnd_Alias NETWORKING**

Make sure it is uncommented

Now scroll down to the bottom of the file and add the following

ADMINS ALL=NETWORKING

Switch to a tty2 with ctrl+alt+f2 and login as peter

As peter try to run the ifconfig command

$/sbin/ifconfig

It should fail

Now run it as sudo

$sudo /sbin/ifconfig

It should work

Open and closed ports

To identify which ports are open on your system you can use many commands

Using the netstat command

#netstat -an

A similar effect can be done with lsof, but this can list what daemons are listening

#lsof -i | grep http

lsof can be used to show what file a process ID has open. In this case /sbin/init with process ID 1

#lsof -p 1

Nmap can be used to show open ports on local and remote machines

For the local machine using a full 3 way handshake

#nmap -sT localhost

For local machine with UDP

#nmap -sU localhost

For a remote machine at an ip address of 172.16.0.5 using a syn scan (send only a syn)

#nmap -sS 172.16.0.5

Finally for a UDP scan of a remote machine

#nmap -sU 172.16.0.5

The ulimits

The ulimit command can set the amount of resources a user can use

To list the current ulimit values

#ulimit -a

To change a value use the switch output in the -a option

To turn off maximum locked memory

#ulimit -l unlimited

Task 2 - Setup Host Security

Services running on a machine are a security risk.

To see a list of daemons and their runlevels at which they run

#chkconfig --list

To disable a service on a running port you have to stop the service

#/etc/init.d/httpd stop

To permanently stop this service from starting automatically

#chkconfig --levels 345 httpd off

To turn it back on

#chkconfig --level 345 httpd on

#/etc/init.d/httpd start

The shadow password system

To unconvert the shadow password system

#pwunconv

Now look in /etc/passwd. You should see the password hashes are now stored in this file

```
#cat /etc/passwd
```

To convert it back use

```
#pwconv
```

Stopping all users from logging in

```
#touch /etc/nologin
```

Switch to the tty2 with ctrl+alt+f2 and try to login as a normal user.

This should fail

Switch back to the X window system and remove the **/etc/nologin**

```
#rm -f /etc/nologin
```

Task 3 - Securing Data with Encryption

Log in as root and make sure you are in your home directory of root

#cd

To start with execute gpg which will create the directories in your home directory

#gpg

Now let's create a public and private keys

#gpg --gen-key

Use 2 DSA and Elgamal

Keysize being 2048

The key does not expire

Enter your First Name and Last Name as the real name

Enter an email address you wish to use

And Enter a comment like Home Email

Select OK

Enter a passphrase to protect the keys

It will then create the keys

You might get a comment on not enough random data to create keys

If this is the case try running something like

#md5sum /dev/sda & in another shell

It will complete and return to a shell prompt

To view the keys created use

#gpg --list-keys

Now export your public key in ascii format, where Real Name is the name you used when creating the key

#gpg -a --export "Real Name" > publickey

Give this file to the people who need to send you encrypted documents

#cp ~/publickey ~peter/

Log in as user **peter** on a **tty2 ctrl+alt+f2**

First you need to import their public key

$gpg --import publickey

You can also download someone's key from a key server

```
$gpg --search-keys --keyserver hkp://subkeys.pgp.net “Richard Millett”
```

To upload your public key to a server

```
$gpg --send-keys --keyserver hkp://subkeys.pgp.net “Richard Millett”
```

To encrypt a file with someone’s public key

Let’s send a copy of **/etc/passwd.** Copy this file into your home directory

```
$cp /etc/passwd ~
$gpg -e --recipient “Real Name” passwd
```

This will create a file called passwd.gpg with the public key created earlier (Change “Real Name” to the name used earlier)

Switch back to the root user and see if you can decrypt the file

```
#cp ~peter/passwd.gpg ~
```

To decrypt at the other end

```
#gpg --output newpasswordfile --d passwd.gpg
```

To view your secret keys on your key ring

```
#gpg --list-sec
```

Switch back to the user peter in tty2 with ctrl+alt+f2

We are going to delete the public key

$gpg --delete-key “Real Name”

Agree to delete the key