INB351/INN351 Practical One (Week One)

Introduction to Linux/Linux Refresher

**Introduction**

**Special Note**

*This practical, like pretty much all practicals this semester, will introduce a number of Unix commands. It is a good idea to learn about the command before you execute it. Especially if you are executing that command with root privileges. Before you use a command, view the man page relating to the command.*

**Environment in S Block Level 5**

All of the practical exercises in this unit will use the virtual machines stored in the lab desktop PCs. You will find the virtual machines under the VMware folder in the Start menu on the Windows 7 operating system. There are 5 virtual PCs that we will use throughout the semester but for this practical we will only be using the Ubuntu-2011 virtual machine.

You will automatically be logged in using the following generic account on the Ubuntu-2011 virtual machine:

Username: vmuser

Password: 90opl;./

When using the other virtual machines you will have to manually login using these credentials.

**Relevant ITB005 Material**

For those who are new to Unix, consider looking at these ITB005 Lab exercises. All ITB005

Lab notes are available on the class Blackboard site.

*•* Lab 1 - Basic Linux Usage

*•* Lab 2 - Directories and Files

*•* Lab 3 - File Manipulation

*•* Lab 4 - Exploring Linux Filesystems

*•* Lab 6 - File Links

**Exercise 1 Getting Help**

Unix provides man pages (manual pages) for practically all installed software. The man page should always be the first point of reference in determining how a program works, and for working out why it doesn’t work. In addition, man pages exist for a lot of the common configuration files as well.

**Q 1.** Browse the man page for man. Read about the -k option in particular

vmuser@gatewaypc:~$ man man

man - an interface to the on-line reference manuals

man -k printf

Search the short descriptions and manual page names for the keyword

printf as regular expression. Print out any matches. Equivalent

to apropos -r printf.

**Q 2.** Using man -k <keyword> determine what command to use to do the following:

1. print the current working directory

man –k print | more

pwd - print name of current/working directory

1. make directories

mkdir - make directories

c) list the directory contents

ls - list directory contents

1. move or rename a file

mv - move (rename) files

1. copy a file or a directory

cp - copy files and directories

f ) remove a file (delete)

remove - remove a file or directory

g) remove a directory which is empty

rmdir - remove empty directories

h) search for files within a directory tree

find - search for files in a directory hierarchy

i) display lines with a file/files containing a particular pattern

grep, egrep, fgrep, rgrep - print lines matching a pattern

**Exercise 2 Getting Around the File System**

**Q 1. Where am I?** On some Unix systems the default prompt will not show the current working directory, or it maybe displayed in a cryptic manner. The pwd command returns the Present Working Directory:

vmuser@gatewaypc:~$ pwd

/home/vmuser

**Q 2. Listing Files.** The ls command is useful for getting a list of files in the current directory, or any specified directory. Look at the man page for ls now:

vmuser@gatewaypc:~$ man ls

After reading the man page, use ls to perform the following tasks using only one command, without piping:

ls - list directory contents

-a, --all do not ignore entries starting with .

-F, --classify append indicator (one of \*/=>@|) to entries

-h, --human-readable with -l, print sizes in human readable format (e.g., 1K 234M 2G)

-l use a long listing format

-r, --reverse reverse order while sorting

-S sort by file size

-t sort by modification time

a) List all files, including hidden files, showing all details

ls –l -a

vmuser@gatewaypc:~$ ls -l -a

total 176

drwxr-xr-x 27 vmuser vmuser 4096 2011-08-03 14:07 .

drwxr-xr-x 3 root root 4096 2011-07-17 22:18 ..

-rw------- 1 vmuser vmuser 2248 2011-08-03 14:51 .bash\_history

-rw-r--r-- 1 vmuser vmuser 220 2011-07-17 22:18 .bash\_logout

-rw-r--r-- 1 vmuser vmuser 3353 2011-07-17 22:18 .bashrc

drwx------ 7 vmuser vmuser 4096 2011-08-03 13:31 .cache

drwxr-xr-x 10 vmuser vmuser 4096 2011-07-19 13:23 .config

drwx------ 3 vmuser vmuser 4096 2011-07-18 22:31 .dbus

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:31 Desktop

-rw-r--r-- 1 vmuser vmuser 63 2011-08-03 13:30 .dmrc

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:31 Documents

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:31 Downloads

-rw------- 1 vmuser vmuser 16 2011-07-18 22:31 .esd\_auth

-rw-r--r-- 1 vmuser vmuser 179 2011-07-17 22:18 examples.desktop

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 23:33 .fontconfig

drwx------ 4 vmuser vmuser 4096 2011-08-03 13:30 .gconf

drwx------ 2 vmuser vmuser 4096 2011-08-05 08:37 .gconfd

-rw-r----- 1 vmuser vmuser 0 2011-07-18 23:24 .gksu.lock

drwx------ 8 vmuser vmuser 4096 2011-07-19 13:24 .gnome2

drwx------ 2 vmuser vmuser 4096 2011-07-19 13:24 .gnome2\_private

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:34 .gstreamer-0.10

-rw-r--r-- 1 vmuser vmuser 142 2011-08-03 13:30 .gtk-bookmarks

dr-x------ 2 vmuser vmuser 0 2011-08-03 13:30 .gvfs

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 hardlinksample

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 hardlinksample2

-rw------- 1 vmuser vmuser 1976 2011-08-03 13:30 .ICEauthority

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:53 .icons

drwxr-xr-x 3 vmuser vmuser 4096 2011-07-18 22:34 .local

drwx------ 4 vmuser vmuser 4096 2011-07-19 13:26 .mozilla

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:31 Music

-rw-r--r-- 1 vmuser vmuser 19 2011-08-03 13:49 mysample1

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:34 .nautilus

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:31 Pictures

-rw-r--r-- 1 vmuser vmuser 675 2011-07-17 22:18 .profile

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:31 Public

drwx------ 2 vmuser vmuser 4096 2011-07-28 12:12 .pulse

-rw------- 1 vmuser vmuser 256 2011-07-18 22:31 .pulse-cookie

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 sample1

-rw-r--r-- 1 vmuser vmuser 17 2011-08-03 14:07 sample2

-rw-r--r-- 1 vmuser vmuser 0 2011-07-18 22:36 .sudo\_as\_admin\_successful

lrwxrwxrwx 1 vmuser vmuser 7 2011-08-03 14:03 symlinksample -> sample2

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:31 Templates

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:53 .themes

drwx------ 4 vmuser vmuser 4096 2011-07-18 22:53 .thumbnails

drwxr-xr-x 2 vmuser vmuser 4096 2011-07-18 22:31 Videos

-rw------- 1 vmuser vmuser 3604 2011-08-05 08:42 .xsession-errors

-rw------- 1 vmuser vmuser 6865 2011-08-01 20:46 .xsession-errors.old

vmuser@gatewaypc:~$

b) List file sizes in ”human readable” format

ls –l –h

vmuser@gatewaypc:~$ ls -l -h

total 56K

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Desktop

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Documents

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Downloads

-rw-r--r-- 1 vmuser vmuser 179 2011-07-17 22:18 examples.desktop

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 hardlinksample

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 hardlinksample2

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Music

-rw-r--r-- 1 vmuser vmuser 19 2011-08-03 13:49 mysample1

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Pictures

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Public

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 sample1

-rw-r--r-- 1 vmuser vmuser 17 2011-08-03 14:07 sample2

lrwxrwxrwx 1 vmuser vmuser 7 2011-08-03 14:03 symlinksample -> sample2

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Templates

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Videos

vmuser@gatewaypc:~$

c) List all files, sizes in human readable format, sorted by file size, *From smallest to biggest*

ls –l –h –S –r

vmuser@gatewaypc:~$ ls -l -h -S -r

total 56K

lrwxrwxrwx 1 vmuser vmuser 7 2011-08-03 14:03 symlinksample -> sample2

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 sample1

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 hardlinksample2

-rw-r--r-- 3 vmuser vmuser 7 2011-08-03 14:00 hardlinksample

-rw-r--r-- 1 vmuser vmuser 17 2011-08-03 14:07 sample2

-rw-r--r-- 1 vmuser vmuser 19 2011-08-03 13:49 mysample1

-rw-r--r-- 1 vmuser vmuser 179 2011-07-17 22:18 examples.desktop

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Videos

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Templates

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Public

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Pictures

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Music

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Downloads

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Documents

drwxr-xr-x 2 vmuser vmuser 4.0K 2011-07-18 22:31 Desktop

vmuser@gatewaypc:~$

d) From your home directory, list all files in /etc/ from oldest to most recent, sorted by *modification time*

ls –l –t –r /etc | more

vmuser@gatewaypc:~$ ls -l -t -r /etc | more

total 1136

-rw-r--r-- 1 root root 2064 2006-11-24 05:33 netscsid.conf

-rw-r--r-- 1 root root 1343 2007-01-10 04:39 wodim.conf

-rw-r--r-- 1 root root 10852 2007-04-28 12:27 gnome-vfs-mime-magic

-rw-r--r-- 1 root root 15752 2009-07-26 01:13 ltrace.conf

-rw-r--r-- 1 root root 645 2009-11-26 18:23 zsh\_command\_not\_found

drwxr-xr-x 2 root root 4096 2010-05-10 00:49 insserv.conf.d

-rw-r--r-- 1 root root 882 2010-05-10 00:49 insserv.conf

drwxr-xr-x 3 root root 4096 2010-05-10 00:49 insserv

drwxr-xr-x 2 root root 4096 2010-05-10 01:44 libpaper.d

-rw-r--r-- 1 root root 624 2010-05-16 21:07 mtools.conf

-rw-r--r-- 1 root root 395 2010-06-20 18:11 anacrontab

-rw-r----- 1 root daemon 144 2010-06-28 05:36 at.deny

-rw-r--r-- 1 root root 2570 2010-08-06 01:57 locale.alias

-rw-r--r-- 1 root root 310 2010-10-15 10:24 updatedb.conf.BeforeVMwareTools

Install

-rw-r--r-- 1 root root 3587 2010-10-15 22:15 lftp.conf

-rw-r--r-- 1 root root 91 2010-10-21 22:47 networks

-rw-r--r-- 1 root root 267 2010-10-21 22:47 legal

-rw-r--r-- 1 root root 92 2010-10-21 22:47 host.conf

-rw-r--r-- 1 root root 12 2010-10-21 22:47 debian\_version

--More--

**Q 3. Working with Directories.** Making, changing, and deleting directories are com- mon tasks that must be performed. Examine the man pages for mkdir and rmdir. There is no specific command to rename a directory or file, however this behaviour can be achieved by moving the directory from one name to another. The mv utility allows for files and directories to be moved.

**Extra Technical Information** The change directory command cd does not have a man page as it is built into the command shell (bash). Information on cd is available in the bash man page. Note, most of what is in the bash man page is not examinable in this unit.

After reading the man pages, do the following tasks:

a) Create a sub directory in your home directory called weekone

b) Change into that sub directory

c) Verify that you are in the weekone directory by displaying the current working directory

d) Verify that the directory is empty using ls

e) Using the .. short cut, change to the parent directory f ) Rename weekone to notweekone

g) Remove the notweekone directory

h) Verify that the notweekone directory has been deleted

vmuser@gatewaypc:~$ mkdir weekone

vmuser@gatewaypc:~$ cd weekone

vmuser@gatewaypc:~/weekone$ pwd

/home/vmuser/weekone

vmuser@gatewaypc:~/weekone$ ls

vmuser@gatewaypc:~/weekone$ cd ..

vmuser@gatewaypc:~$ mv weekone notweekone

vmuser@gatewaypc:~$ rmdir notweekone

vmuser@gatewaypc:~$ ls

Desktop Downloads Music Public Videos

Documents examples.desktop Pictures Templates

vmuser@gatewaypc:~$

**Exercise 3 Dealing With Files**

**Q 1. Viewing Files.** It is a common task to view the contents of a plain text file. A

number of programs are capable of doing this:

*•* cat short for concatenate

cat - concatenate files and print on the standard output

-n, --number

number all output lines

*•* more a program to display a page of text at a time

more — file perusal filter for crt viewing

*•* less a program to display a page of text at a time, with more features than more

less - opposite of more

Less is a program similar to more (1), but which allows backward move‐

ment in the file as well as forward movement. Also, less does not have

to read the entire input file before starting, so with large input

files it starts up faster than text editors like vi (1).

*•* head displays a number of lines from the start (head) of a file

head - output the first part of files

-n, --lines=[-]K

print the first K lines instead of the first 10; with the lead‐

ing `-', print all but the last K lines of each file

*•* tail displays a number of lines from the end (tail) of a file

tail - output the last part of files

-n, --lines=K

output the last K lines, instead of the last 10; or use -n +K to

output lines starting with the Kth

-f, --follow[={name|descriptor}]

output appended data as the file grows; -f, --follow, and --fol‐

low=descriptor are equivalent

Look at the man page for these five tools. They are all useful for specific tasks, but most of the time you will want to use less.

For example, to view the /etc/passwd file you would do:

vmuser@gatewaypc:~$ less /etc/passwd

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh

contd...

After reading the man pages try and answer these questions:

1. Name two differences between more and less?

Less is a program similar to more (1), but which allows backward movement in the file as well as forward movement. Also, less does not have to read the entire input file before starting, so with large input files it starts up faster than text editors like vi (1).

1. How would you display the first 5, and only 5, lines of a file?

head -5 filename

1. How would you display the last 100 lines of a file, and any following lines that get appended?

tail -100 -f filename

**Q 2. Deleting Files.** To delete files in Linux, use the rm command. Look at the man page for rm then:

a) Create a file using vim in your home directory

b) Use ls to verify the file was created

c) Delete the file using rm

d) Again, using ls verify the file was deleted

vmuser@gatewaypc:~$ vim weekone

vmuser@gatewaypc:~$ ls

Desktop Downloads Music Public testfile weekone

Documents examples.desktop Pictures Templates Videos

vmuser@gatewaypc:~$ rm weekone

vmuser@gatewaypc:~$ ls

Desktop Downloads Music Public testfile

Documents examples.desktop Pictures Templates Videos

vmuser@gatewaypc:~$

**Q 3. Copying and Moving Files.** To copy or move files in Linux, use the cp and mv commands. Browse the man pages for these commands and then create a file, copy it and then move it.

vmuser@gatewaypc:~$ cp testfile testfile2

vmuser@gatewaypc:~$ ls

Desktop Downloads Music Public testfile Videos

Documents examples.desktop Pictures Templates testfile2

vmuser@gatewaypc:~$ mv testfile2 testfile3

vmuser@gatewaypc:~$ ls

Desktop Downloads Music Public testfile Videos

Documents examples.desktop Pictures Templates testfile3

vmuser@gatewaypc:~$

**Q 4. Text Searching.** Searching for text within a file is another common task that must be performed. A number of tools exist for text manipulation and searching which will be used in later weeks. However, for simple searching within a group of files, the grep utility is sufficient.

Begin by browsing the man page for grep. At this time it is not necessary to know about extended expressions or regular expressions.

After browsing the man page, do the following:

a) Display all lines that contain the word *db* in the file /etc/nsswitch.conf

grep db /etc/nsswitch.conf

b) Find lines containing *monthly* in /etc/logrotate.conf, making sure to print 2 lines before and 2 lines after the matching line

grep -A 2 -B 2 monthly /etc/logrotate.conf

1. Print the number of lines containing *false* in /etc/passwd

grep -c false /etc/passwd

a)

vmuser@gatewaypc:~$ grep db /etc/nsswitch.conf

protocols: db files

services: db files

ethers: db files

rpc: db files

vmuser@gatewaypc:~$

b)

vmuser@gatewaypc:~$ grep -A 2 -B 2 monthly /etc/logrotate.conf

/var/log/wtmp {

missingok

monthly

create 0664 root utmp

rotate 1

--

/var/log/btmp {

missingok

monthly

create 0660 root utmp

rotate 1

vmuser@gatewaypc:~$

c)

vmuser@gatewaypc:~$ grep -c false /etc/passwd

11

vmuser@gatewaypc:~$

**Exercise 4 Input/Output Redirection**

Reference: <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html>

The previous exercises aimed to introduce a number of Unix command line tools and show off some of the functionality offered. However, the true power of the command line is visible when multiple commands are tied together using input/output redirection. Redirection is possible because most Unix command line tools (by default) read input from Standard Input (stdin) and direct the output to Standard Output (stdout). A second output source called Standard Error (stderr) also exists.

There are two types of redirection, redirection to a file or device (which is just a file), or redirection to another application. This is made possible by one of these operators:

> Write stdout to a file. Overwriting the file if it exists, creating the file if it doesn’t exist

>> Concatenate (append) stdout to an existing file. If file does not exist, create a new file

2> Write output of stderr to a file. Overwriting the file if it exists, creating the file if it doesn’t exist

| Pipe the stdout of one command to stdin of another. This is one of the most useful redirection operators

Additional examples are shown at the above URL.

**Q 1.** a) Use redirection to save a copy of the directory listing of your home directory to a file named homedirlist

ls > homedirlist

b) Use piping along with cat and less to concatenate /etc/passwd and /etc/group together and display the output one page at a time. *Caution. Do not do this as root in case you overwrite the system files by accident*

cat /etc/passwd /etc/group | less

a)

vmuser@gatewaypc:~$ ls > homedirlist

vmuser@gatewaypc:~$ cat homedirlist

Desktop

Documents

Downloads

examples.desktop

homedirlist

Music

Pictures

Public

Templates

testfile

testfile3

Videos

vmuser@gatewaypc:~$

b)

vmuser@gatewaypc:~$ cat /etc/passwd /etc/group | less

**Exercise 5 Combining it All Together**

The power of the Unix command line is realised by combining commands together. This is called piping. Piping allows you to redirect the output of one command to the input of another. So instead of displaying to the screen, it becomes the input for the second command.

To pipe the output of one application to the input of another we use the pipe operator (|). For example, to pipe the output of tail -n 100 (which is bigger than the screen) into less we would do:

vmuser@gatewaypc:~$ tail -n 100 /var/log/messages | less

**Q 1.** For this exercise we want to get a list of all the files in /usr/lib that have the word *lib* somewhere in their name. However, there are many files that contain *lib* so we want to display one page of text at a time.

The three commands we will use are: ls, grep and less.

Use your knowledge of ls, grep and less along with piping to produce the desired result.

Note: Unix provides a much more powerful file searching utility, find. find will be introduced in a later week.

vmuser@gatewaypc:~$ ls /usr/lib | grep lib | less

**Exercise 6 Editing Files**

The most common command line based text editor in Linux is vi or its improved cousin, vim. There are alternatives to vi such as nano (which is installed in the S block level 5 environment) or emacs. While vi can be difficult to use initally it is a very powerful text editor and is likely to be installed on any Unix like operating system, whereas some of the other alternatives may not.

As this week’s exercise is quite short, you should spend some time going through some of the many vi tutorials available online, or the quick introduction in the ITB005 Lab 3 exercises.

Done

**Exercise 7 Symbolic Links (symlinks)**

Most Unix file systems allow the creation of file ’links’. These are commonly referred to as *symlinks* even though file links can be either ’hard’ or ’symbolic’. In most cases a symbolic link is the type of link you want to use, only use hard links when necessary.

A symlink allows you to create one or more pointers to a single real file or directory. When an application or process tries to access a symlink, the filesystem automatically fol- lows the link to the real file and proceeds as normal. The utility of symlinks will become more apparent as you work through the semester.

Try the following simple exercise to demonstrate how symlinks work:

**Q 1.** Take a look at the man page for ln

ln - make links between files

-s, --symbolic

make symbolic links instead of hard links

**Q 2.** Using a text editor such as vi, create a file called realfile.txt in your home directory containing any piece of text you like. Exit from vi ensuring that you save the file.

vmuser@gatewaypc:~$ vi realfile.txt

**Q3.** Use the ls command to verify the file was created.

vmuser@gatewaypc:~$ ls

Desktop examples.desktop Pictures Templates Videos

Documents homedirlist Public test1

Downloads Music realfile.txt testfile

**Q 4.** Using the ln tool, create a symlink to realfile.txt called linkfile.txt

vmuser@gatewaypc:~$ ln -s realfile.txt linkfile.txt

**Q 5.** Use the ls command to verify the symlink was created.

vmuser@gatewaypc:~$ ls

Desktop examples.desktop Music realfile.txt testfile

Documents homedirlist Pictures Templates Videos

Downloads linkfile.txt Public test1

**Q 6.** Use the cat or less command to view the contents of linkfile.txt

vmuser@gatewaypc:~$ cat linkfile.txt

this is a real file

and this is line 2

vmuser@gatewaypc:~$

**Q 7.** The contents should be the same as what you saved in realfile.txt

vmuser@gatewaypc:~$ cat realfile.txt

this is a real file

and this is line 2

**Exercise 8 Command Line Time Savers**

As the Unix command line has been around for 30 years, there has been ample time to improve the user interface and the user experience. Two of the most common and useful features are Tab Completion and the Bash History.

*Part I — Tab Completion*

If you need to type in a particular file name or path, and that filename or path exists, you can usually partially type some of the name, then press the *tab* key to autocomplete the name. For example:

1. The command that you would use to configure a network interface is ifconfig

2. At a command prompt try typing ifc and then press *tab*

Now, suppose you wish to view the contents of the file

/etc/udev/rules.d/70-persistent-net.rules using the more command. Normally you would use the command:

vmuser@gatewaypc:~$ more /etc/udev/rules.d/70-persistent-net.rules

However, you can use tab completion to save some typing. Press *tab* at the end of the end of each of the following lines (but do not press *enter* until the end):

1. mor

2. /e

3. ud

4. ru

5. 70

Notice that the generated command line is missing net.rules from the end, this is because more than one file matches the search string of 70. Press *tab* twice and bash will reveal all of the files that match. Finish the command line by typing n and then pressing *tab*.

*more /etc/udev/rules.d/70-persistent-net.rules*

*Part II — Bash History*

bash automatically keeps a record of every single command you type, internally, this history is stored in the file ~/.bash\_history (~/ is a short way of referring to the current users home directory). From the command line you can scroll through the history by using the *up* and *down* arrows.

1. By now you should have typed a number of commands and therefore, have some commands stored in the history. View the contents of the bash history file.

2. Use the arrows to cycle through the history

Being able to use the arrows to go back to recent commands is very useful, consider the situation if you made a mistake typing the command cd tiles instead of cd files. Rather than retyping the entire command, you could just press the *up* arrow once, then correct the mistake.

However, as the history grows in size, sometimes it is easier to be able to search the command line history. You can tell bash to search this history by pressing *Control + r*

1. Generate some command history by doing each of these commands (don’t worry about the actual commands or output for now). Remember, use tab completion to save your fingers!

(a) dmesg

(b) ifconfig

(c) tail /var/log/messages

(d) head /etc/passwd

(e) tail /etc/group

(f ) ls /etc

(g) cat /proc/cpuinfo

2. Now we want to re-run the tail /var/log/messages command without retyping the command, and without scrolling up using the arrow keys

3. Press the *Control* and *r* keys, notice that the bash prompt has changed

4. Type /var

5. You should notice the command appear, if it doesn’t, type some more of the command

6. When the command you want fully appears, press *enter*

tail /var/log/messages

**Exercise 9 Homework**

Learn about and experiment more with vim. A good handy cheat sheet is available here:

<http://www.fprintf.net/vimCheatSheet.html>

done