

Unix/Linux Preparation Course

June 27, 2010

Exercises: Using Commands

1. Log in as the *pacnog* user:

At the graphical login window log in using

```
username: pacnog
password: <given in class>
```

2. Open a terminal window

Under Applications menu open the Accessories sub-menu and then choose Terminal. Once you have a terminal window open you can proceed with the exercises.

3. Become the *root* user

At the command prompt type the following command:

```
$ su -
```

You will need to enter the password for the *root* user given in class.

Now that you are *root* the command prompt will change. We indicate this using the “#” symbol.

4. Install the *sudo* package (*already done, but you can do this for practice*)

The *sudo* package allows you to run system commands as a regular user. To install *sudo* do the following:

```
# apt-get install sudo
```

Now we must do a bit of configuration. Feel free to ask your instructor what this really does. Note the spaces in the following command

```
# visudo
```

Go down the file until you find these two lines:

```
# Members of the admin group may gain root privileges
#%admin ALL=(ALL) ALL
```

We want to uncomment the second line to look like this (see steps below):

```
%admin ALL=(ALL) ALL
```

To do this delete the “#” character, press ctrl-x to exit, say “Y” to save the file and then press <ENTER> to choose the default filename listed.

Now go back to being a “regular” user (i.e. pacnog):

```
# exit
```

5. View files:

Use `ls` to list files:

```
$ cd [Moves you to your home directory]
$ ls [Show files in the pacnog home dir]
```

Do you see anything? Try this instead:

```
$ ls -lah [Shows all files, including hidden files]
```

What's inside one of these files? (If there is no `.bashrc` file use another “.” file)

```
$ cat .bashrc
```

Try:

```
$ less .bashrc
```

Press “q” to get out of the `less` display.

Now try:

```
$ clear
$ cat .bashrc
```

If you don't understand what `cat`, `clear` or `less` do, then type:

```
$ man cat
$ man clear
$ man less
```

And, now try:

```
$ more .bashrc
```

6. Working with the command prompt:

You can recover previous commands by using the up-arrow and down-arrow keys. Give this a try now.

Alternately, try typing this command:

```
$ history
```

If you wish to execute one of the commands in the list you saw type:

```
$!nn
```

Where “nn” is the number of the command in the history list. This is useful if you want to run a past command that was long and/or complicated.

Command completion:

With the bash shell you can auto-complete commands using the tab key. This means, if you type part of a command, once you have a unique string if you press the TAB key the command will complete. If you press the TAB key twice you'll see all your available options. Your instructor will demonstrate this, but give it a try by doing:

```
$ hist <TAB>
$ del <TAB><TAB>
$ rm <TAB><TAB> [Include the space after the “rm”]
```

7. Working with pipes:

We saw an example of using pipes when we sorted the contents of our /sbin directory during the presentation. What if you wanted to have this information available in a file and sorted?

```
$ cd [Make sure we are in our home dir]
$ ls /sbin | sort > sbin.txt
```

Now view the contents of what is in sbin.txt to verify that this worked. Do you remember how to do this? If not, go back to exercise 5.

8. Finding text strings:

Use the command `grep` to print lines matching a pattern in a data stream (such as a file). For example, view the entry for the *pacnog* account in the system `passwd` file:

```
$ sudo grep pacnog /etc/passwd
```

Note the use of the `sudo` command. The file `/etc/passwd` is restricted with r/o access for for the *root* user. To view the file you need to elevate your privilege level while executing the `grep` command. This is what `sudo` does. You will be prompted for a password – enter in the password for the *pacnog* user.

You should see something like:

```
pacnog:*:1000:1000:PacNOG 7 User:/home/pacnog:/bin/bash
```

The previous items above are:

```
userid:passwd:uid:gid:Name:HomeDir:LoginShell
```

`grep` is often used with a pipe to reduce the number of results. For instance:

```
$ history | grep ls
```

Will display your previous use of the `ls` command from exercise 5.

9. Editing the command line revisited:

It is particularly useful to realize that you can edit a command just as you would a line of text in a file. For instance, you can:

- Use your back-arrow and forward-arrow keys to change text in a command.
- Use the **Home** and **End** keys to go to the start and the end of a command.
- Note: you *do not* need to go to the end of a command before pressing <ENTER> to execute the command.
- You can use the `history` command with `grep` to find a previous command. You can copy and paste this command, then edit it to make adjustments. For long commands this can save considerable time.
- Alternatively you can use the reverse-search feature of bash:

- 1.) Press `ctrl-r`.
- 2.) type the term you are searching for.
- 3.) Press `ctrl-r` to cycle through all occurrences of the term in your history.
- 4.) Press the right or left-arrow, HOME or END key to start editing the command.

Let's give some of these editing rules a try. Instead of searching for the *pacnog* user in the `/etc/passwd` file we'll search for the *root* user.

- 1.) Press “`ctrl-r`”
- 2.) type “`pacnog`”
- 3.) *If necessary* press `ctrl-r` again until you see the original command you used to search for the *pacnog* user in the `/etc/password` file (`sudo grep pacnog /etc/passwd`).
- 4.) Use the arrows keys to move in the command and change *pacnog* to *root*. You should now have the following on your command line:

```
$ sudo grep root /etc/passwd
```

With your cursor positioned over just past the “t” in “root” – Press <ENTER> to execute the command. You should see:

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

That's it. You can practice repeating the above with other parts of commands you have already executed (`ls`, `grep`, `cat`, etc.).

10. Copy and pasting commands:

A nice feature in Unix is the built-in automatic copy buffer (think “automatic clipboard”). As soon as you highlight text it is available in your copy buffer. You *do not* need to use the `ctrl-c`, `ctrl-v` keyboard combination to copy and paste text.

After you have highlighted text, then you can place your cursor where you want to paste the highlighted text and press the *middle* mouse button to do the text paste.

Give this a try:

```
$ history | grep sudo
```

Locate the previous command from exercise 9 and highlight just that command. Highlight the command text (not the number next to it).

Now press the middle mouse key. The command should be pasted to your prompt.

Did this work? Now you could edit the command or just press enter to execute it.

This is very useful if you have two terminals open and want to use text from one terminal in a different terminal, or if you wish to copy resulting text from a command in one terminal in to a file that's open in another terminal – etc.

11. Software Installation

If you are logged in as the *root* user please do the following:

```
# exit
```

To become a normal user, like *pacnog*. Your prompt should change to include a “\$” sign.

```
$
```

Figure out what package installs the “*lynx*” text-based web browser

```
$ apt-cache search lynx
```

There are a lot of potential packages listed, but this one looks the most likely (and obvious):

```
lynx - Text-mode WWW Browser (transitional package)
```

Let’s install the *lynx* package. To do this do

```
$ sudo apt-get install lynx
```

Once the *lynx* package is installed we can try using it.

```
$ lynx nsrc.org
```

Press “q” to exit *lynx*.

Q.) What do you think is happening?

A.) You are viewing the home page for the Network Startup Resource Center (NSRC) using a text-based web browser called *lynx*.

Try viewing our classroom web pages:

```
$ lynx noc
```