Linux System Administration and IP Services
AfNOG 2012

## Linux Commands

# Notes

* Commands preceded with "$" imply that you should execute the command as a general user - not as root.
* Commands preceded with "#" imply that you should be working as root with "sudo"
* Commands with more specific command lines (e.g. "RTR-GW>" or "mysql>") imply that you are executing commands on remote equipment, or within another program.

1.      Log in as the sysadm user using ssh

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

2.      Become the root user

At the command prompt type the following command:

```
        $ sudo -s
```

        Enter your own password when prompted

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

You are now the super user - be careful!

Ok, exit the root account:

```
        # exit
        $
```

3. Look at the network configuration of your host

```
        $ cat /etc/network/interfaces
```

Notice that configuration of your host is done using DHCP.
"cat" is for "concatenate" and is one way to view what is in a file.

4.      List files:

Use ls to list files:

```
        $ cd                            [go to your home directory]
        $ ls
```

Do you see anything? Try this instead:

```
        $ ls -lah
```

What's inside one of these files?

```
        $ cat .profile
        $ less .profile
        $ clear
```

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

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

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

5.      Working with the command prompt:

You can recall 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”]
```

6.      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
        $ ls /sbin | sort > sbin.txt
```

Now view the contents of what is in sbin.txt to verify that this worked.

```
        $ less sbin.txt
```

Press the "q" key to quit viewing the contents.

7.      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 sysadm account in the system passwd file:

```
        $ grep sysadm /etc/passwd
```

You should see something like:

```
sysadm:x:1000:1000:System Administrator,,,:/home/sysadm:/bin/bash
```

The previous items above are:

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

grep is often used with a pipe to FILTER the output of commands. For instance:

```
        $ history | grep ls
```

Will display your previous use of the ls command from exercise 2.

8.      Stopping and starting a service

In a web browser go to your machine's home page:

```
        http://pcN.ws.nsrc.org/
```

You should see something like "It Works!" on the page. Now, let's stop the web server (Apache) that is installed on your virtual machine. To do this you can do:

```
        $ sudo service apache2 stop
```

Now reload the web page for your machine. It should indicate that no web server was found. Now let's start the service again:

```
        $ sudo service apache2 start
```

You can see if a service is running by typing:

```
        $ sudo service apache2 status
```

If a process ID is displayed, then the service is running, but our next exercise will show you another way to verify this.

9. Finding and stopping processes

If you wish to find something that is running and then stop it you can use the "ps" (process) command with "grep" and "kill". Let's do this by opening two connections to your virtual machine.

a. Using SSH open two terminal connections to your Linux server (user: sysadm machine: pcN.ws.nsrc.org)
b. One you have opened two terminals go in to one terminal and type:

```
        $ tail -f /var/log/syslog
```

This will let you look at the end of the syslog log file in real time. If events take place that are logged you will see them as they happen. Now, in your other terminal let's look for this process:

```
        $ ps auxwww | grep tail
```

The "auxwww" are options to the ps (process) command. The options mean display all process running that belong to you and to other users and provide information about who owns what process. The three "www"'s mean display the entire command line regardless of how long it is and wrap it in your window.

You will likely see something like this:

```
root      6903  0.0  0.0   7200   612 pts/0    S+   03:28   0:00 tail -f /var/log/syslog
root      6986  0.0  0.0   9388   924 pts/1    S+   03:28   0:00 grep --color=auto tail
```

You could press "CTRL-C" in the terminal window where the tail command is running, or to stop the process right now you can use the kill command. You need to replace the Process ID (PID) with the process ID number of the tail command running on your machine. In this example the number is "6903". At the command prompt type:

```
        $ kill NNNN
```

Where NNNN is the PID of your tail process. Once you do this return to the other terminal screen. The "tail -f" process should now have exited and you should see something like:

```
        Terminated
        sysadmt@pcN:~$
```