IP/Unix Preparation Course
May 29, 2010

Exercises: Using Commands

0. **Connect to your assigned Unix or Linux machine.**

   If you are using a Desktop machine these are the steps from the initial desktop login prompt:

   1. Click on the "User Afnog 2011" entry.
   2. In the "Password:" box enter in the password for the "afnog" user (given in class).
   3. Open a terminal window. Go to Applications menu ➔ Accessories ➔ Terminal
   4. In the terminal window start an ssh session to your virual machine by typing:

      $ ssh afnog@10.10.0.NNN

   Where "NNN" is the address of the machine assigned to you. Your instructor will give you this in class.

   If you are using a laptop you will need to either have available or install an SSH client program. Once you do this, then you will open a connection to your assigned machine and continue with Step 1.

   Your class instructor will assist you to use SSH the first time.

1. **Log in as the *afnog* user:**

   username: *afnog*
   password: <*given in class*>

2. **View files:**

   You are now logged in as the *afnog* user and your initial location is in the *afnog* user's home directory located at /usr/home/afnog.

   Use `ls` to list files:

   ```
   $ cd                    [go to your home directory]
   $ ls                    [probably nothing is visible]
   ```

   Do you see anything? Try this instead:

   ```
   $ ls -lah
   ```

   View the contents of a file? (If there is no .mailrc file use another "." file)

   ```
   $ cat .mailrc
   ```

   Try:

   ```
   $ less .mailrc
   ```

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

Now try:

```
$ clear
$ cat .mailrc
```

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

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

And, now try:

```
$ more .mailrc
```

3. **Working with the command prompt:**

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

Next, try typing the 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 likely demonstrated this, but give it a try by doing:

```
$ hist<TAB>
$ del /<TAB><TAB>          [please, don’t delete anything! ;-)]
$ rm <TAB><TAB>            [Include the space after the “rm”]
```

4. **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. Do you remember how to do this? If not, go back to exercise 2.

5. **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 *afnog* account in the system passwd file:

   ```
   $ sudo grep afnog /etc/passwd
   ```

   Note the use of the `sudo` command. The file `/etc/passwd` is restricted with r/o access given just to 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 *afnog* user.

   You should see something like:

   ```
   afnog:*:1001:0:Admin User:/home/afnog:/usr/local/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 2.

6. **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:

     1. Press CTRL-r

2. Type “sbin”
3. Press the End key
4. Delete the text “> sbin.txt” and replace it with “| more”

You should now have the command:

```
$ ls /sbin/ | sort | more
```

With your cursor just past the “more”. Press <ENTER> to execute the command. You should the contents of the /sbin directory appear on your screen in a single column in ascending alphabetical order. You can press “q” to exit the more command, or space to see additional pages.

7. **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 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 sbin
```

Locate the previous command from exercise 7 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.

8. **Software Installation (Optional exercise. May take too long to complete)**

You need to become *root* for this exercise. To do this do:

```
$ su -
```

Your prompt should change to include the “#” sign.

```
#
```

Install some tools to help us install packages more simply.

```
# pkg_add –r portmaster
```

Tell the *root* user’s default shell (“sh”) to reload it’s path in memory by typing:

```
# rehash
```

Figure out what port(s) installs the "*lynx"* text-based web browser

```
# cd /usr/ports
# make search name="lynx" | more
```

You will see quote a few possible choices, but read the first line of the package description that says "Port:" and, then, the directory listing just below that to see how you can actually tell Portmaster to install the particular port. In our case we have two reasonable choices:

```
Port: lynx-2.8.7.1_1,1
/usr/ports/www/lynx

Port: lynx-2.8.8.d7
/usr/ports/www/lynx-current
```

Let's install just the lynx package. To do this do:

```
# portmaster -P www/lynx
```

Once lynx finishes installing you need to tell the shell for the root user to reload it's path in memory (again) by typing. You don't need to do this for the *afnog* user as its default shell is the *bash* shell.

```
# rehash
```

And, now you can actually use the lynx text-based web browser to view a page from your terminal session:

```
$ 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) in a text-based web browser called *lynx*.