Ubuntu System Administration

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.

Find out what's installed

Log on to your machine using SSH as the user specified in class.

Once you are logged in, take a look at all the packages installed on your system:

```
$ dpkg --get-selections
```

All installed packages fly by on the screen. Let's slow that down:

```
$ dpkg --get-selections | less
```

The “less” command lets you quickly search text. Is the “openssh-server” server installed on your machines? (It should be if you are logged in :)

```
Type "/openssh" and press <ENTER>
```

You should see something like:

```
openssh-client                                  install
openssh-server                                  install
```

with the “openssh” text highlighted. Press “q” to exit the less screen.

Another way to see packages is:

```
dpkg --list | less
```

Try it!

What version of “openssh-server” is installed?

```
$ apt-cache policy openssh-server
```

Or, you could also say:

```
$ dpkg --list openssh-server
```

Find out if a package is available to be installed

You have a local cache of all packages available to be installed from the Ubuntu package repositories. You can search this cache using the “apt-cache” command. Before you can use apt-cache the first time you need to update your local cache. Let's do this now (we did this for you when setting up your machine):

```
$ sudo apt-get update
```

Once this completes we can search for available packages. Let's see if the “ipcalc” package is available in our Ubuntu repositories:

```
$ apt-cache search ipcalc
```

It looks like there are three packages matching the name “ipcalc”. Try typing:

```
$ sudo apt-get install ipcalc
[sudo] password for sysadm: .... <- your password
```

Then try:

```
$ ipcalc 41.93.45.101/24
```

This is very useful! We'll talk more about what all this means later today or tomorrow.

NOTE: If we ask you to run a command on the system and you are told it isn't available, you can use this process to install it.

Stopping and starting services

The scripts to run services on your machine are located in /etc/init.d/. By default, when Ubuntu installs a package the startup scripts for the package are run and the package is configured to automatically run at system startup.

Try viewing the status of the ssh server, stopping and starting the server and reloading the server's configuration file (/etc/ssh/sshd_config):

```
$ service ssh help
```

You are shown the commands you can perform on the ssh service.

Try to view the status of the ssh server:

```
$ sudo service ssh status
```

Since we are connected using ssh we cannot stop this service. If we did, then you would lose your connection and need to go to your machine's console to manually restart the service.

Use the top command

The top command let's us see the status of our system at a quick glance. To use top simply do:

```
$ top
```

The item at the top of list of running processes is the process using the most CPU resources.

Open a new SSH connection to your PC. In that window type:

```
$ ls -lahR /
```

Now in the other window where top is running you should start to see the “ls” process listed using some amount of your total CPU.

At the top of the top window you'll see something like:

```
top - 03:17:03 up  1:47,  2 users,  load average: 0.51, 0.19, 0.09
Tasks:  79 total,   2 running,  77 sleeping,   0 stopped,   0 zombie
Cpu(s):  4.9%us, 10.9%sy,  0.0%ni,  3.6%id, 79.6%wa,  1.0%hi,  0.0%si,
0.0%st
Mem:    508924k total,   491968k used,    16956k free,    59052k buffers
Swap:   905208k total,     4584k used,   900624k free,   128712k cached
```

This is a good, quick way to see how much RAM, Virtual memory, CPU, total running processes, etc. that your machine has, and is using.

You can adjust the output of top as it is running. Exit from top by typing “q” and then do:

```
$ man top
```

Now run top again and change what it is displaying interactively.

HINT: You can also type **h** while running top to get quick summary of commands

All the information in top is part of a dynamic file system located in /proc. As an example do the following:

```
$ cd /proc
$ ls
```

The numbered directories correspond to actual Process IDs of processes that are running. Look at the file meminfo:

```
$ less meminfo
```

Remember: space bar to go to the next screen of output.

Note that it includes your total RAM. Top uses this file to get this information. Same for cpuinfo, loadavg, uptime, etc.

If you want to know what command was executed to start a number process you can type (for instance):

```
 $ less /proc/1/cmdline
```

You'll see that the first process started on the system is init.

Viewing your log files in real time

In a window connected to your pc type:

```
$ sudo tail -f /var/log/auth.log
```

Now open a second window to your PC and login using ssh.

What do you see in the first window?

In the second window try:

```
sudo -s
```

What do you see in the first window?

Now go back to being the **sysadm** user by typing:

```
exit
```

What do you see in the first window?

Now close your second ssh session using:

```
exit
```

What do you see in the first window?

This method of looking at log files can be very useful if you are trying to debug problems. We'll use it later.