COMP 150 Lab 7 Cron and Packages

Part 1: Cron

Make a crontab file in your home directory

NOTE: A cronfile has two parts.

The first part is the environment SHELL, HOME, MAILTO,PATH etc.

a. Add these four variables with appropriate values at the top of your cronfile.

b. Add the command line that prints keeps a running log in your home directory of date.
   i. Run it every Friday in 10 minute intervals between 8pm and midnight.
   ii. Redirect the output to a file named datelog in your home directory.
   iii. Be sure to include the full path for date. Use **whereis** to find it.

TAKE A SCREENSHOT: of the cronfile.

Install the crontab file

a. Use the command **crontab *cronfile*** to do this. *"cronfile" is the name you gave your file.*

list the file with **crontab -l**

TAKE A SCREENSHOT: show the command and the listing. This listing is coming from /var/spool/cron and verifies that the file is installed as a cron tab.

b. edit the file with **crontab -e.** This will edit the file in var/spool/cron, not the file in your directory.

   Change the day you run date to Tuesday.

TAKE A SCREENSHOT: Show the system response to your edit. You have changed the file in /var/spool/cron but you have not changed the file in your own directory.

c. Update your directory file with the following:
   crontab -l > ***yourcronfile in your directory***
   Note that you need to use '>' to redirect and replace.

TAKE A SCREENSHOT of datelog once you start getting data. If you don't have cron running, you can email this to me or update your lab once you do.

Part 2: Packages

Now it's time to shift gears and move to package management. In CentOS, most software packages are managed via the **`yum`** command:

1. You can list all available software packages with: **`yum list`**
2. Of course, you often use `grep` to search for particular packages. For example, let's say we want to install Wireshark. First, we'll check if it's available:
   `yum list | grep -i wireshark`
3. This will only list packages whose name matches `wireshark` (case insensitive due to the `-i`).
4. When you run the above, you should see two packages: `wireshark` and `wireshark-gnome`.
5. At first, it may not be clear what the difference in the two packages is, so we can use the **`yum info`** command.
6. Run: **`yum info wireshark`**
7. The description indicates that this package includes libraries and command line utilities.
8. Now run: **`yum info wireshark-gnome`**
9. The description indicates that the package contains desktop integration for Gnome (the GUI interface).
10. So, we'll install the GUI version, which should also install the command line package.
11. To do that use the `yum install` command:
    `sudo yum install wireshark-gnome`
12. `yum` will prompt you to ensure you want to proceed with installing 3 packages:
    `wireshark-gnome`,
    `wireshark`, *and*
    `libsmi`.
13. Enter "`y`" to actually install the packages.

This general procedure (`yum list`, `yum info`, `yum install`) is how you usually go about finding and installing packages with `yum`. Once the packages have been downloaded and installed, Wireshark will be available to run on your VM:

1. You should see Wireshark in the GUI menus under `Applications->Internet`.
2. You can also start it via the terminal by running: `wireshark`.
3. In either case, it will prompt you for a password to run in privileged mode (required to actually capture packets).

4. **TAKE A SCREENSHOT** of the Wireshark window in your VM.

Another common `yum` task is to update the system with newer versions of packages as they become available:

1. First, you can check if there are any updates with: `sudo yum check-update`
2. Chances are you don't have any updates and so it won't list any packages.
3. If packages are listed with updates available, then you can update all of them with: `sudo yum update`

Finally, you sometimes need to remove packages if you no longer need them installed on your system:

1. First check if the package is actually installed with **`yum list`**.
2. For example, let's remove the Evolution mail program, so first check if it's installed with: **`yum list | grep -i evolution`**.
3. You should see a number of Evolution related packages, but the base package is simply "`evolution`" which will show up first in the list. More importantly, you see that it is indeed installed because it's listed explicitly as "`installed`" in the far right column.
4. To remove it: `sudo` **`yum erase evolution`**
5. `yum` will prompt you to be sure you want to remove the two listed packages.
6. Say "y" and they will be removed from your system.

Sometimes packages you want to install are not available via `yum` because the CentOS development team hasn't included it in the online repositories. There are many reasons for this, but often it's simply that there hasn't been enough demand for developers to dedicate time to the package to add it to the repositories officially. When this happens, you have to find the package yourself either as an RPM file ready to install or as the original source code which you then have to compile yourself. We are going to go through both of these processes now.

First, let's assume you were excited by the ccrypt package. Always verify that it's not available via `yum` first:

1. Run: **`yum list | grep -i ccrypt`**
2. In this case, there are no packages listed, so we assume it's not in the repositories.

Occasionally you have to fall back on compiling programs yourself:

1. Do a Google/Yahoo/Bing/whatever search for the package and locate it on the web.
2. For example, ccrypt is hosted on sourceforce: http://sourceforge.net/projects/ccrypt/
3. You will see a button to download the ccrypt pack. It will be tarred (many files collected into one) and gzipped.

4. Unzip the file: **gunzip ccrypt-1.10.tar.gz.** You will see a file **ccrypt-1.10.tar** in you Download directory
5. Form the directory ccrypt-1.10
    a. **./configure**
    b. **make**
    c. **make check**
    d. **sudo make install**
    e. **make clean**

6. To verify it is indeed installed, try running the `ccrypt` command: `ccrypt --help`
7. **TAKE A SCREENSHOT** of the above `ccrypt --help` command. You should get a help message explaining how to use the `ccrypt` program.

Keeping your bash scripting skills in shape:

For some extra credit, you can enhance the `add_new_users.sh` script in one of two ways (or both if you're feeling ambitious).

First, ensure that usernames don't collide with existing user names. For example, if you already have an account named `philipf` and a new user is named "Philip Fry" then the new user can't be `philipf` also. Instead you have to add a digit to the end of the username: `philipf1`. Of course, there could also be a `philipf1`, in which case it must be `philipf2`, and so on. To do this, you'll need to look through the `passwd` file with `grep` to check if the default username is already in use and if so, add/update the digit. This check will have to be done in a loop that continues until a username not in use is found.

Second, the random password generation we used is not terribly good. For example, it is possible (though unlikely) that you could generate a password of "`password`" or other Very Bad passwords. So, enhance the script by checking that the random password is Good Enough before proceeding. For this script, assume that means that there is at least one lower case letter, one upper case letter, and one digit. That means you'll have to examine the password and if you don't find at least one of each then generate a new one. Of course, you'll have to loop until you get a good password.

Once you're done enhancing the script and have tested it thoroughly, copy and paste the script file into your submission document and **TAKE A SCREENSHOT** of the output of the script using a sample user list. For the first enhancement be sure you are adding a user that will exercise the extended usernames.