CS 279 - Week 14 Lab Exercise

Deadline

Due by the end of lab on 2022-12-01.

How to submit

Submit the files specified below on https://canvas.humboldt.edu.

Purpose

To try logging into a Linux desktop, practice writing a `case` statement, try out `crontab` a bit, and try out some of the other commands discussed this week.

Important notes

- Work in PAIRS for this lab exercise:
  - two people at one computer,
  - one typing (driver),
  - one saying what to type (navigator),
  - both discussing along the way!

  When done, the driver should e-mail the files to the navigator, so BOTH of you can EACH submit them.
- Assume, for all `bash` scripts in this course, that the following are required:
  - Start each script *(EXCEPT for a script containing JUST Bash functions)* with the line that is considered good style (and is a CS 279 course requirement), that specifies that this script should be executed using the `bash` shell
  - After a blank line, put in one or more **comments** including at least the name of the shell script, your names, and its last modified date
  - And follow these comments with a blank line.

Problem 1

At:

http://nrs-projects.humboldt.edu/~st10/f22cs279/desktop-help/a-few-tips.php

...you will find some quickly-adapted-from-another-thing notes for logging into **vlinux.humboldt.edu**, so you can experience a Linux desktop a bit.

Use these to log into vlinux.humboldt.edu, and when you have done so, let me know, so I can come and admire your vlinux desktop and check off that you have completed this problem.

It would be ideal if you now use the vlinux desktop for the problems below, just to get to try using it. You can ssh to nrs-projects from here if you would like, also.

Problem 2

Consider the Bash `case` statement. Hopefully, you can see that it could be very useful for handling a command-line-based "menu" of user options.

Write a little bash shell script `menu-play.sh` that outputs to the screen a menu of at least 5 options, asks the user to enter an option, and then handles that option; it will repeatedly do this until the user enters the "quit" option from that menu. (As always, make sure that it includes both of your names!)

What can the options be? One should be a "quit" option, as noted above, but otherwise, you get to choose -- maybe they are different styles of greetings (1: formal, 2: informal, 3: rude, etc.), or perhaps different UNIX/Linux commands the script will do for you (1: see current directory, 2: see current time, etc.).

Note that if something looks like it would make a nice function, feel free to do so! (I would certainly make printing out the menu options be a function, for example, since you will probably print it initially and then within a loop...)

Submit your resulting `menu-play.sh`.

Problem 3

Now, let's try our hands at `crontab`.

On nrs-projects, the default editor for `crontab` is `vi`. You can change it for today's lab by changing the `EDITOR` environment variable to another editor -- for example:

```
export EDITOR=/usr/bin/nano
```

You create/edit your `crontab` file using the command:

```
crontab -e
```

(-e for edit).

You insert a `crontab` entry in this file for each desired job to be scheduled. As discussed in class, it contains 6 fields, separated by blanks:

1. minute (0-59)
2. hour (0-23)
3. day of the month (1-31)
4. month of the year (1-12)
5. day of the week (0-6, 0=Sunday)
6. command to execute

Remember:

- you can have comment lines, BUT they must be either ALL blank, or have a # in column 1
- if you put an asterisk in one of the fields 1-5, it represents ALL valid values
- a field can contain a number, a comma-separated list, or a range separated by a dash
- field 6 consists of a command line optionally followed by lines that the command is to read from standard input.
- I've found that you need to use absolute pathnames in `crontab` entries, both for the commands and for

any files being redirected to -- beware!

So -- create a crontab file with at least one comment and at least two crontab entries:

- the comment should include both of your names
- one entry should be a very simple crontab entry that fires every minute, appending the result of the date command to a file crontab-play.txt. Be sure to use absolute pathnames for both the command and the file name!
- the other can run a bash shell script of your choice, that meets the following criteria:
  - use at least one non-* value for fields 1-5, and make sure it will run at least once during lab, since we are going to remove these crontab entries at the end of lab
  - make sure the shell script redirects its output so that it appends it to crontab-play.txt
- make sure the crontab file ends with a newline

Make sure that you crontab is indeed doing what you want/expect.

Then -- this crontab file is not one in a typical place! So, BEFORE you remove its contents below, copy and paste its contents into a lab14-3-crontab.txt file, so you can submit it. Or, simply list its contents into that file, with the command:

```
crontab -l > lab14-3-crontab.txt
```

When you are happy with how your crontab entries are performing AND you have made a copy of your crontab file's contents in lab14-3-crontab.txt, then, because we are NOT going to keep this just running and running, REMOVE your crontab file by using the command:

```
crontab -r
```

...and show me that you've done so by running the following commands:

```
date > final-ck.txt; crontab -l 2>> final-ck.txt
```

Submit your resulting crontab-play.txt, lab14-3-crontab.txt, a copy of the shell script used for your second crontab entry, and final-ck.txt.

Problem 4

To give you a chance to practice some of the commands we've discussed recently, write a Bash script lab14-play.sh that includes the following(feeling free to add blank lines or underlines for readability as desired):

- echo a message including both of your names
- echo a message indicating that you are about to run the date command, and then run the date command, with at least **two** format descriptors of your choice
- echo a message indicating that you are about to run the cal command, and then run the cal command with a month and year of your choice
- echo a message indicating that you are about to run the uname command, and then run the uname command with the -a option
- echo a message indicating that you are about to run the du command, and then run du twice on a non-empty directory of your choice. Run it once with the -a option and once with the -s option.

- `echo` a message indicating that you are about to run the `uptime` command, and then run the `uptime` command
- `echo` a message indicating you are bout to try out the `time` command, and then run the `time` command with a Linux command or Bash script of your choice.
- feel free to try out more if you would like! 8-)

Run `lab14-play.sh`, redirecting the output into a file `lab14-out.txt`

Submit your resulting versions of `lab14-play.sh` and `lab14-out.txt.`

Submit these files to Canvas:

- `menu-play.sh`

- `crontab-play.txt`
- `lab14-3-crontab.txt`
- a copy of the shell script used for your second `crontab` entry
- `final-ck.txt`

- `lab14-play.sh`
- `lab14-out.txt`