

CS 279 - Week 14 Lab Exercise - 11-28-12

Deadline

Due by the end of lab on 11-28-12.

How to submit

Submit your files for this lab using `~st10/279submit` on **nrs-labs**, with a homework number of 94

PLEASE NOTE -- since you will be doing MOST of this lab on **nrs-projects**, that does indeed mean that you need to `sftp` your lab files from **nrs-projects** to **nrs-labs** in order to submit them! Think of it as a little more `sftp` practice.

Purpose

Practice with the `bash case` statement, `RCS`, and `crontab` (with a little `sftp` on the side)

Important notes

- You are expected to work in **pairs** for this lab; this means **two** students at **one** computer and **one** keyboard, one typing, and the other suggesting what to type. Both are expected to be engaged and involved in what is going on, and **each** file that you create should have **both** of your names in it.

Students who do not work in pairs may not receive credit for the lab exercise (although if an odd number of students requires it, there may be one authorized trio -- the same guidelines apply).

- If you have a question during lab, and I am helping another pair, add one or both of your names to the "Next:" list on the board, and I will get to you as soon as I can.
- `RCS` is not on **nrs-labs**! But it is on **nrs-projects**, so you can practice with it there.
- Your answers may be posted to the course Moodle site.

Problems

Problem 1

Do this problem on **nrs-projects**, so that we can use it to play with `RCS` as well as `case` statements.

Consider the `bash case` statement that we discussed in class yesterday. 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...)

ALSO: Create an RCS revision group for your script as well; check-in at least THREE versions as you are developing your code.

Submit your resulting latest version of `menu-play.sh` as well as your `menu-play.sh,v`. (I have modified `~st10/279submit` to accept `,v` files along with the other suffix-types.) (Remember, you need to `sftp` these files to `nrs-labs` to submit them.)

Problem 2

Now, let's try our hands at `crontab`. This problem, too, must be done on **nrs-projects**, since you are permitted to make your own `crontab` file on that account.

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

```
crontab -e
```

(-e for edit). Note that the default editor is `vi`, so you will also be getting some more `vi` practice, if this isn't already your usual text editor.

- note that you CAN change that default editor -- change the value of the `EDITOR` environment variable, being sure to `export` it, and using the full pathname of your desired editor. Remember that the `which` command can be used to get the full pathname of a command... 8-)

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

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 commend 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 output to a file that you can submit as part of this lab exercise
- 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 lab14-2-crontab.txt file, so you can submit it. Or, simply list its contents into that file, with the command:

```
crontab -l > lab14-2-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 lab14-2-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, lab14-2-crontab.txt, a copy of the shell script used for your second crontab entry, a copy of the file resulting from your second crontab entry, and final-ck.txt. (Remember, you need to sftp these files to nrs-labs to submit them.)