CS 279 - Homework 3

Deadline:

Due by 11:59 pm on **FRIDAY**, September 12.

How to submit:

Submit your files using `~ah270/279submit` on nrs-labs, with a homework number of `3`, by the deadline shown above.

Purpose

To think about filename expansion, job control, and UNIX files and directories.

Important notes:

- If I don't specify which mode -- octal or symbolic -- to use in specifying a file's permissions, then you may use either mode, your choice.
- It is possible that your answers may be collected and posted to the course Moodle site.

The Problems:

Problem 1:

Consider a directory containing the following files:

`lab1.txt`	`lab2.txt`	`lab3.txt`	`OLDlab1.txt`
`OLDlab4.txt`	`ancestors.pl`	`save-it`	`slab-msrmts.txt`
`rules.rkt`	`lab45.rkt`	`lab754.pl`	`lab9.cpp`

In a file named `hw3-prob1.txt`, put your name, and then your answers to each of the following.

1 part a

Write an `ls` command, that uses only a SINGLE argument including file expansion wildcard character(s) after its option, that would give a long-listing (including permissions) of only those files with a suffix of `.txt`

1 part b

It turns out that `save-it` is a directory. Write a command, that uses only a SINGLE argument including file expansion wildcard character(s) before the directory name `save-it`, that would copy ONLY those files with a suffix of `.txt` whose names before this suffix consist only of `lab` followed

by a single character.

1 part c

Give the names of the files that would be moved into the parent directory by the following command:

```
mv *lab* ..
```

Submit your resulting `hw3-prob1.txt`.

Problem 2:

Consider the following output of a `jobs` command:

```
[1]   Running                 actions.sh &
[2]   Stopped                 emacs -nw hw3-prob1.txt
[3]+  Stopped                 vi hw3-prob3.txt
[4]   Stopped                 ~ah270/279submit
[5]-  Stopped                 nano brilliant.sh
[6]   Stopped                 doIt.sh
```

In a file named `hw3-prob2.txt`, put your name, and then your answers to each of the following.

2 part a

Assume that the above `jobs` command was just done. Give a command that would cause the `emacs` process shown above to become the foreground process.

2 part b

Assume that the above `jobs` command was just done. What would become the foreground process as a result of the following command?

```
%%
```

2 part c

Assume that `doIt.sh` does not ask for standard input nor does it output to standard output. Write a command *or* describe a sequence of actions that would change it from being `Stopped` to actually running in the background.

2 part d

Assume that the above `jobs` command was just done. Write a command that would kill the `nano` process above.

2 part e

Assume that the above `jobs` command was just done. Now the following command is done:

```
another-task.sh &
```

Assume that `another-task.sh` is an executable bash shell script, that does not ask for standard input nor does it output to standard output, and that it takes quite a long time to complete. Type out the output that would result if the `jobs` command is now done again, before this script has had a chance to complete.

Submit your resulting `hw3-prob2.txt` .

Problem 3:

Consider the directory `/Users/cbrown` . It contains a subdirectory `football` that contains:

```
game1-score.txt     f12-schedule.txt          past-seasons
```

..where `past-seasons` is itself a directory that contains:

```
f11-schedule.txt fl1-scores   f10-schedule.txt f10-scores
```

In a file named `hw3-prob3.txt`, put your name, and then your answers to each of the following.

3 part a

Give the **absolute** pathname for the `f11-schedule.txt` file described above.

3 part b

Assume that you are within the directory `/Users/cbrown` (and that you have appropriate permissions). Write a *single* command, using a **relative** pathname, that would give the long listing (including permissions) for the file `f10-schedule.txt` described above.

3 part c

Assume that you are within the directory `/Users/cbrown` (and that you have appropriate permissions). Write a *single* command that would move file `f12-schedule.txt` described above into the directory `past-seasons` described above.

3 part d

Assume that you are within the directory `past-seasons` described above. Write a *single* command -- *without* changing your current working directory, and *without* using an absolute pathname -- that will give the long listing (including permissions) for the directory `football` described above. (Note that we want the permissions for the directory itself, *not* for its contents.)

Submit your resulting `hw3-prob3.txt` .

Problem 4:

Write a `bash` shell script `current-status.sh` that:

- contains a comment containing your name, and the last-modified date
- prints to the screen a message of your choice saying here is info about the current working directory,
- ...followed by the absolute pathname of the current working directory,
- ...followed by the long listing (including permissions) of the current working directory (for the directory itself, NOT for its contents)
- prints to the screen a message of your choice saying that it is about to list the contents of the current working directory
- ...followed by that list of the contents of the current working directory
- prints to the screen a message of your choice saying it is about to list the current jobs, if any,
- ...followed by the list of current jobs
- prints to the screen a message of your choice saying it is about to list all processes you own, regardless of the shell they are in,
- ...followed by the list of such processes

Also perform at least the following test of `current-status.sh`:

- do this test within a directory containing at least 3 non-directory files -- I'll call this directory 1, but it can be named whatever you like.
- start up at least three background processes while within directory 1 -- stopped processes are fine. (For example, you could start up several calls to the `man` command, either starting them in the background or starting them and then typing ^Z to put them in the background.)
- start up another `ssh` window -- let's call this shell 2. Start up something within that shell 2 -- it could be a `nano` command, or a background process (stopped or running), your choice -- but keep it running until you complete this test.
- back in your original shell, in directory 1, run `current-status.sh` from the *current* shell (instead of starting up a new little shell) with the help of the `source` command, redirecting the results to the file `current-status-test.txt`:

  ```
  source current-status.sh > current-status-test.txt
  ```

Submit your resulting `current-status.sh` and `current-status-test.txt`