CS 279 - Homework 10

Deadline

11:59 pm on **Friday, December 9**

Purpose

To practice more with the Bash `case` statement, `crontab`, and several other recently-discussed commands.

How to submit

Submit your files to the course Canvas site.

NOTE: While I list the separate files you need to submit for each problem below, I am going to set up Canvas to *also* accept `.zip` files.

That is,

- you can submit each file to Canvas,
- OR, if you prefer, you may compress your files to be submitted into a single `.zip` file and submit that `.zip` file to Canvas.

Important notes

Assume, for all `bash` scripts in this course, that the following are required:

- Start each script 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 name, and its last modified date
- And follow these comments with a blank line.

Problem 1

Write a Bash shell script `get-details.sh` that **appropriately uses a `case` statement** such that, for each file in the current directory, it FIRST outputs its name on one line, and THEN, starting on the next line, follows that with additional output based on the file's name as described below.

Note that you are permitted to add additional spacing and "underlines" or borders as you wish.

- for a file name ending in `.txt`, it then outputs the result of the `wc` command for that file.
- for a file name ending in `.sh`, it then outputs each line in that file (if any) that happens to contain a command-line argument
- for a file name ending in `.gz`, it then outputs the results of calling `gzip -l` on that file (this gives a lovely summary of its compression)
- for a file name ending in `.tar`, it then outputs the result of calling `tar tf` on that file, so that the archive's contents are displayed.
- otherwise, (and you *do* want an `if` statement within this default case), if the file name is a directory, it then outputs a message including the number of files in that directory, else it outputs the long listing for that file.

Submit your resulting `get-details.sh`.

Problem 2

In a file `hw10-2.txt`, put your name, and then your answers for each of the following.

2 part a

Write a `date` command using at least two format descriptors of your choice.

Then, run this `date` command, and paste in the result.

2 part b

Write a `cal` command with at least one argument of your choice.

Then, run this `cal` command, and paste in the result.

2 part c

Oh dear -- you discover, using the `uptime` command, that the system load is quite high. Give the command you could type to see, in real-time, which processes are using the most CPU until you type `q` or control-d. (Don't make this harder than it is -- I just want to get this command into this homework, even if trivially!)

2 part d

You are writing an interactive Bash script in which, for dramatic effect, you'd like the script to echo something to standard output, pause for about 4 seconds, and then echo something else (such as the punch line of a joke).

Write the command you could put between those two `echo` commands to accomplish this.

2 part e

You have an executable Bash script `experiment.sh` in your current working directory, that does not happen to need any command-line arguments.

You'd like to run this, and then see the real time, user time, and system time it took to run that script.

Write the command to do this.

Submit your resulting `hw10-2.txt`.

Problem 3

In a file `hw10-3.txt`, put your name, and then your answers for each of the following.

3 part a

Consider this `crontab` entry:

```
1 2 3,5 6-8 4  /home/abc1/glarble.sh
```

What will this cause to happen, and when/how often?

3 part b

Consider this `crontab` entry:

```
30 21 * * 5 /usr/bin/find /home/def2 -name "*~" -print >> /home/def2/check-these.txt
```

What will this cause to happen, and when/how often?

3 part c

Write a crontab entry that will run a script `/home/ghi3/monthly.sh` at 12:05 am on the first day of each month.

3 part d

A system administrator with username `jkl4` wants to gather some system performance statistics snapshots.

Write a crontab entry that will append the result of the `uptime` command to file `/home/jkl4/fall-aft-stats.txt` every Monday and Wednesday from September to December at 5:00 pm each day

3 part e

As we discovered in the Week 14 Lab:

- the results of the `time` command go to standard error, not the standard output -- this is so it can be possible to time piped commands!
- and you need to enclose the `time` command and the command you wish to time in parentheses, and *then* redirect to standard error.

That is,

```
time ls 2>> looky.txt      #  does NOT redirect ls's execution times to looky.txt
(time ls) 2>> looky.txt    #  DOES redirect ls's execution times to looky.txt
```

That said...

...assume that it is currently 4 pm on an unspecified date. User `mno5` want to arrange to have a shell script, `/home/mno5/big-big-task.sh`, run one time, at 2:00 am tomorrow morning, such that the amount of time it takes to run is also appended to a file `/home/mno5/task-stats.txt`. Write up a command that `mno5` could do now to set up this task to be done then.

3 part f

User `mno5` wants to run another shell script, `/home/mno5/low-priority.sh`, in the background, with a priority 19 lower than the norm. Write a command that `mno5` could do to make this happen.

Submit your resulting `hw10-3.txt`.

Submit your resulting files:

- `get-details.sh`
- `hw10-2.txt`
- `hw10-3.txt`