CS 279 - Week 10 Lab Exercise - 10-28-2014

Deadline

Due by the end of lab on 10-28-2014.

How to submit

Submit your files for this lab using `~ah270/279submit` on nrs-labs, with a homework number of `90`

Purpose

Practice with customizing your prompt, setting your PATH, and setting command aliases.

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.
- Your answers may be posted to the course Moodle site.

Problems

Problem 0

Create a shell script `lab10src.sh` in which you will be playing around with mucking with your prompt, setting up some command aliases, and adding to your command `PATH`.

Include both of your names and today's date in a descriptive comment within this shell script.

And, make this executable and readable, for testing purposes.

Problem 1

Now look over the different special characters you can use in customizing a bash command prompt, available at:

`http://www.gnu.org/software/bash/manual/bashref.html#Printing-a-Prompt`

(and this link is also set up under today's in-class examples section already, for your convenience).

Save your current prompt's setup with:

```
SAVE_PS1=$PS1
```

Now play around with resetting `PS1` and see the effects of some of these different special characters.

Decide on a prompt that:

- uses at least two special characters **BESIDES** those used in the default `PS1` on nrs-labs, `\u`, `\h`, and `\W`
- includes at least two non-special characters of your choice

...and set `PS1` to this within `lab10src.sh`. Remember that you can run a shell script "within" the current session (and not in its own subshell) with the `source` command -- and so test that your prompt is appearing as you like by running:

```
source lab10src.sh
```

(and remember that you can "restore" your prompt to the default using:

```
PS1=$SAVE_PS1
```

...assuming, of course, that you saved it at the beginning of your session as instructed.)

Interesting/entertaining prompts MAY be posted to the course Moodle site.

Problem 2

We practiced adding to one's `PATH` in an earlier assignment (Homework 5, Problem 1, part f). But this is also something one would typically do in a `.bashrc` file, so let's practice with this in your `lab10src.sh`.

Remember that you generally want to ADD to your `PATH`, not remove the system directories to check for UNIX/Linux commands already there -- so, always START with:

```
PATH=$PATH:...  # and then add : followed by
                #     additional-directory-of-your-choice for
                #     EACH additional directory you wish to add
                #     to your PATH
```

Two typical additions one might make to your path is to make your own `bin` directory, and add that to your `PATH`, and to add the current working directory, to make running new shells more convenient.

Create a `bin` directory in your home directory, if you have not already, and create a shell script of your choice there, being sure to make it executable and readable. Then `cd` back to the directory where your `lab10src.sh` is being developed.

Then, in `lab10src.sh`:

- echo the current/initial `PATH` value
- add these two directories (your new `bin` directory and the current working directory character) to your `PATH`, and

- echo the new/resulting `PATH` value.

Test your latest version of `lab10src.sh` by:

- running `source lab10src.sh`
- check out the pre- and post-change values of `PATH`
- try running the shell you put into your `bin` directory from this directory -- it should now work, if you've successfully reset your `PATH`, and
- try running a shell script from your current directory -- you now should not need `./` before its name, to do so.

Problem 3

We also practiced setting up command aliases in Homework 5, Problem 1, in parts g and h. But, again, this is something one would typically do in a `.bashrc` file, so let's practice with this in your `lab10src.sh`.

Recall: you can set a command alias using:

```
alias newcomm="meaning of newcomm"
```

First, create an alias for `rm` so that when you type `rm` you actually get `rm -i` (since every UNIX/Linux user should consider doing this, even if they then decide not to.)

Then, create two more command aliases of your choice.

Add these to `lab10src.sh` and test that your aliases are working by running:

```
source lab10src.sh
```

...and then trying out your new command aliases.

Interesting/entertaining/useful aliases MAY be posted to the course Moodle site.

Submit your resulting `lab10src.sh`.

Problem 4

In a file `lab10-4.sh`, put commands for each of the following:

- echo that this is the starting state (followed by a border, blank line, etc. if you would like).
- Do a `pwd` command to note where you are when the script starts.
- Choose a file -- put an `ls -l` command showing the initial state of this file.
- Choose a directory BESIDES this directory -- put two commands, `ls -ld` and then `ls -l`, showing the initial state of this directory
- echo that you are about to use `gzip` and `gunzip` (followed by a border, blank line, etc. if you would like).
- Use `gzip` to compress your chosen file, followed by an `ls -l` command to show that you now have a gzipped result

- use `gunzip` to uncompress that file you just compressed, followed by an `ls -l` command to show that it is now gunzipped
- echo that you are about to use `tar` (followed by a border, blank line, etc. if you would like).
- use `tar` to create a `.tar` file from the chosen directory, followed by an `ls -l` showing the tarred file resulting
- change to another directory of your choice, and run `pwd` to show you are elsewhere
- use `tar` to unpack that tar file back into directory form in this different directory, followed by `pwd`, an `ls -ld` and a `ls -l` for the now-untarred directory copy, showing a copy of it is now in this different directory.

Run your `lab10-4.sh`, redirecting its output into `lab10-4-test.txt`, and submit your resulting `lab10-4.sh` and `lab10-4-test.txt`.