

CS 279 - Week 10 Lab Exercise

Deadline

Due by the end of lab on 2022-10-27.

How to submit

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

Purpose

To try out some `tee` commands, to create and use a Bash array, and to try out using the `BASH_REMATCH` array.

Important notes

- This exercise assumes that whoever is serving as navigator has a `bin` directory in their home directory, and that this `bin` directory has been added to their `PATH` environment variable (as was set up during the Week 7 Lab Exercise).
- 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 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.

Lab Exercise setup

- use `ssh` to connect to the one of your accounts on `nrs-projects.humboldt.edu`
- make and protect a directory `279lab10` using the commands:

```
mkdir 279lab10  
chmod 700 279lab10
```

- go into that directory using:

```
cd 279lab10
```

Problem 1

In a file named 279lab10-prob1.txt, put:

- your names
- your answers to the following (except for 1 part f, for which you'll submit the specified file).

1 part a

- Go to the driver's ~/bin directory.
 - If there are not already at least three Bash scripts there, copy several into this directory.
- Write a grep command whose output will be just the names of files in the current directory that contain the line:

```
#!/bin/bash
```

- Hint: you will need to escape the ! in your regular expression.
- Hint: the Week 8 Lecture 2 posted notes include several useful grep options

1 part b

- Now write a command that uses the tee command to capture the piped output of your grep command from 1 part a into the file script-list.txt before piping it onward to a wc command whose output will be the number of lines piped to it.

1 part c

- Write the output of your command from 1 part b when you run it in the driver's ~/bin directory.

1 part d

Fun Fact: if you call wc with more than one file, it lists the desired count(s) as well as the file name.

- Write a command whose output will be the number of lines in each of the files in the current directory.

1 part e

- Now write a command that used the tee command to capture the piped output of your wc command from 1 part d into a file line-sizes.txt before piping it to the sort command (which will then display these in order of least number of lines to the most number of lines).
 - YIKES -- on nrs-projects' version of Bash, you need to use sort -n to get this sorted order!

1 part f

- Run your command from 1 part e in the driver's ~/bin directory, redirecting its output into a file sorted-order.txt.

Submit your file 279lab10-prob1.txt as well as the files script-list.txt, line-sizes.txt, and sorted-order.txt.

Problem 2

Write a Bash script `lab-array` that meets the following requirements:

- Create an array which you initialize to contain at least 6 items, such that:
 - at least one of the items includes at least one blank in its value
 - at least one of the items has an index that is at least 3 greater than the actual number of elements in that array
- Echo to the screen a single expression whose value is the number of elements in your array.
- Echo to the screen a single expression whose value is all of your array's contents (there are two reasonable expressions for doing this).
- Echo to the screen a single expression whose value is all of your array's indexes.
- Write a for-loop that will echo each element in your array to the screen, such that each element is displayed on its own line (even those that contain one or more blanks).
- (and you may add additional statements doing more with your array as you would like).

Run your resulting script, redirecting its output into a file `lab-array-demo.txt`.

Submit your resulting files `lab-array` and `lab-array-demo.txt`.

Problem 3

In a file named `279lab10-prob3.txt`, put:

- your names
- your answers to the following (except for 3 part c and 3 part d, for which you'll submit the specified files).

3 part a

Consider the ERE:

```
for\ (.+)\ \ in\ (.+)\$
```

In general, which lines within a Bash shell script should this match?

3 part b

Consider this first line of an `if` statement:

```
if [[ $next_line =~ for\ (.+)\ \ in\ (.+)\$ ]]
```

In general, for lines that match this ERE, what will be in `${BASH_REMATCH[1]}` ?

In general, for lines that match this ERE, what will be in `${BASH_REMATCH[2]}` ?

3 part c

Write a Bash script `grab-for-info` that expects one command-line argument, expected to be a filename, and tries to read all of the lines from that file looking for lines that match the pattern in 3 part b.

If found, this should echo to the screen descriptive messages including the value of `${BASH_REMATCH[1]}` and `${BASH_REMATCH[2]}` for that matching line, followed by echoing a

blank line.

3 part d

In your current working directory, create a copy of one of the Bash scripts `array-play` (from Week 10 Lecture 1), `quoted-play` (from Week 10 Lecture 2), or `linux-journal-ex` (the Week 10 lecture 2 completed version).

Run your script `grab-for-info` on one of those files, redirecting the result into a file `grab-for-demo.txt`.

Submit your file `279lab10-prob3.txt` as well as the files `grab-for-info` and `grab-for-demo.txt`.

Submit these files to Canvas:

- `279lab10-prob1.txt`
- `script-list.txt`
- `line-sizes.txt`
- `sorted-order.txt`

- `lab-array`
- `lab-array-demo.txt`

- `279lab10-prob3.txt`
- `grab-for-info`
- `grab-for-demo.txt`

BOTH of you should then submit copies of these problems' files to Canvas for this lab exercise.

Once both of you have submitted these lab exercise files, you may leave lab if you wish. Or, you can ask questions, read the course text, etc. But note that questions about today's lab exercise will get first priority.