

CS 279 - Homework 6

Deadline

11:59 pm on Friday, October 28

Purpose

To think about more BRE options and some ERE options, to play around a bit with `cmp` and `diff`, and to write some more Bash shell scripts.

How to submit

You will complete **Problems 1 and 2** on the course Canvas site.

For the rest of the problems, you will create several files and then submit those 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 - 10 points

Problem 1 is correctly answering the "HW 6 - Problem 1 - Short-answer questions on more BRE options" on the course Canvas site.

Problem 2 - 10 points

Problem 2 is correctly answering the "HW 6 - Problem 2 - Short-answer questions on some ERE options" on the course Canvas site.

Problem 3

Consider: what is a regular expression that will match a blank line in a file?

Create a shell script `strip-blank-lines` or `strip-blank-lines.sh` that meets the following specifications:

- it expects exactly one command line argument, expected to be a regular, readable file -- it should complain descriptively and exit with a non-zero exit status if this is not the case
- then, it creates a file `stripped-` followed by the name of the input file that contains the same contents as

the input file EXCEPT with any blank lines stripped out

- (or, if you prefer, it reads all of the lines from the given file, and appends to the new file ONLY non-blank lines from the original input file)

For example, if called with argument `pet.txt`, then, it would create output file `stripped-pet.txt` that contains `pet.txt`'s contents MINUS any blank lines.

Submit your resulting `strip-blank-lines` or `strip-blank-lines.sh`.

Problem 4

Create a file `prob4-play.txt`, making sure that it contains at least 6 lines, at least 3 of which are empty lines that occur before the last non-empty line.

Then, in a file `hw6-4.txt`, include:

- your name
- the part you are giving an answer for
- your answer for each of the following.

4 part a

Run the command (using `strip-blank-lines.sh` if that's what you named your script):

```
./strip-blank-lines prob4-play.txt
```

Then run:

```
cmp prob4-play.txt stripped-prob4-play.txt  
echo $?
```

Type as your answer to this question what is echoed to the screen here.

4 part b

Now run:

```
cmp prob4-play.txt prob4-play.txt  
echo $?
```

Type as your answer to this question what is echoed to the screen here.

4 part c

Now run:

```
diff prob4-play.txt stripped-prob4-play.txt
```

Paste in the output of this command as your answer to this question.

4 part d

Now run:

```
diff stripped-prob4-play.txt prob4-play.txt
```

Paste in the output of this command as your answer to this question.

Submit your resulting files `prob4-play.txt`, `stripped-prob4-play.txt`, and `hw6-4.txt`.

Problem 5

I think there are several possible ways to approach the following shell script, based on some of the topics we have discussed recently. I am curious to see which you decide to use!

Create a shell script `pair-lines` or `pair-lines.sh` that meets the following specifications:

- it expects exactly two command line arguments:
 - if given more or less than that, it should complain descriptively and exit with a non-zero exit status
 - the first is expected to be a regular, readable file -- it should complain descriptively and exit with a non-zero exit status if this is not the case
 - the second is expected to be a string of interest
- Your shell script should print to the screen all lines in the given file that contain at least TWO instances of the given string of interest.
- Optionally, it can end by printing the number of lines that contained at least TWO instances of the given string of interest.

Submit your resulting `pair-lines` or `pair-lines.sh`.

Submit your resulting files:

- `strip-blank-lines` or `strip-blank-lines.sh`
- `prob4-play.txt`, `stripped-prob4-play.txt`, and `hw6-4.txt`
- `pair-lines` or `pair-lines.sh`