CIS 345/545 – Homework 3

In this assignment you will write several small Linux shell scripts. Follow the steps given below carefully, recording the results as you go through them. Each script must print your name when it first starts.

Passing parameters to shell scripts from the command line:

```
#!/bin/csh
# Display arguments
echo Written by YOUR NAME
echo Program name: $0
set n=$#argv
echo There are $n arguments
set i=1
while ($i <= $n)
   echo $i. $argv[i]
   @ i++
   end
```

Part 1: Write individual shell scripts to do the following:

findtext.s Given a string and a file name as parameters, if any lines in the file contain the string, this program displays (once) the file name and then the appropriate lines from the file with line numbers. For example, **`findtext.s be hamlet.txt`** would show:

hamlet.txt
1. To be or not to be- that is the question:
9. Devoutly to be wish'd. To die- to sleep.
15. For who would bear the whips and scorns of time,
21. With a bare bodkin? Who would these fardels bear,
26. And makes us rather bear those ills we have

right.s Displays the 3 integers it is passed as parameters and prints either "is a right triangle" or "is not a right triangle" as may be appropriate. The quotation marks are not printed. For example, **`right.s 3 4 5`** would print:
3 4 5 is a right triangle
while **`right.s 4 5 6`** would print:
4 5 6 is not a right triangle

count.s Displays the number of characters, words, lines, and file name of the file(s) given as parameters. For this program, a word is something that is surrounded by white space. For example, **`count.s speech.txt`** might print:
speech.txt
2345 characters
514 words
62 lines
I would suggest using the Linux command **`wc`** for this script.

Mark each shell script file as being executable (be sure to include the proper **#!/bin/csh** line at the beginning of each shell script as well). You may not write any C programs to be executed by your shell scripts; they may only use shell commands and standard Linux utilities. Two of the scripts will probable be a single line that runs a Linux filter, while the third will be written with just shell script operations.

In a directory that contains the files from lab 2, run each shell script as follows, capturing the output with the Linux **`script`** command.

```
ls -l >tempfile
cat tempfile
findtext.s .s tempfile
right.s 4 3 5
count.s *.c
right.s 3 7 23
```

Part 2: Multifunction Utility Script

Write a new shell script called mu.s that acts as a front-end command line parser, calling the findtext.s, right.s, and count.s scripts as directed. The command line arguments given to mu.s are arranged in groups, each beginning with -f, -r, or -c (which selects either findtext.s, right.s, or count.s function to be used), followed by the normal parameter(s) for that function.

For example, the command line:
`mu.s -r 3 4 5 -f STAND_ALONE right.c`

would perform the right triangle check on sides 3, 4, and 5, and then display the lines in right.c (with line numbers) that contain the string STAND_ALONE. If an invalid - argument is given (such as -x), display an error message with bad - argument and its parameter(s) and then continue with next - argument.

Test program with the following commands, which should be recorded using the Linux script command.

```
ls -l >tempfile
cat tempfile
mu.s -f .s tempfile -r 5 4 3 -x 1.234 alpha -c *.s -r 3 7 23
```

Turn in the following:
Since there are a lot of parts to this assignment, it is especially important that you organize your work so it is easy to find what I will be looking for. Organize things in the order of the steps and the list below.

Line numbered listings of mu.s, findtext.s, right.s, and count.s
(Using cat -n or pr -n are two ways of producing a line numbered listing)
The script captured from part 1
The script captured from part 2

See the instructions on the next page for submission. Do not submit this assignment to Blackboard!!!

Submission:

1. Each student submits one copy of each item listed on the previous page.

2. Create a folder and name it [first name]_[last name]_h3 ($ mkdir mary_jones_h3)

3. Copy all your source code to the above folder (source code only, no binary files!).

4. Copy all your scripts captured from both parts to the above folder.

5. scp the folder (e.g., mary_jones_h3) to your account on grail (if it's not already there).

6. ssh log in grail, go to the parent directory of the folder you created, and run the following: (suppose that your folder is "mary_jones_h3")

turnin22 proj3@cis545j mary_jones_h3

You should see a list of files followed by a message like the following:

You are about to turnin X files for [XKB] for proj3 to cis545j

* * * Do you want to continue?

Answer y or yes to submit or answer n or no if you do not want to submit.

If you answered y or yes, you should now see ab list of the files being submitted, followed by:

* * * TURNIN of proj3 to cis545j COMPLETE! * * *

7. You can submit more than one time. Your most recent submission will always automatically overwrite the previous one. Do note, however, that if your latest submission is sent after the deadline, it may lose points for being late, even if your earlier submission was on time.