Programming Languages: Project #4 - Bash Script

SWE3006 Programming Languages - Fall 2019

Due date: December 13 (Fri) 11:59pm

1 Objective

In this assignment, you will write a bash script grade.sh that "grades" students' programming assignments. The basic idea is that you are working as a part time TA and you were asked to grade a lab assignment. Your professor gave you a set of students' C source codes and you need to write a script that compiles and runs each of them one at a time, examining the output to see if it matches the expected output.

2 Details

To get started, copy students.tar.gz file from /home/swe3006/proj4. This file contains a set of source codes to grade. Extract this file to your proj4 directory with the following tar command.

```
$ tar xzvf students.tar.gz
```

It will decompress files in your current working directory

Student files are named as students IDs and `output.txt` contains the expected output of compiled programs. You will run grade.sh script in the directory where students' source codes are stored.

To read students' files from the current directory, you may want to look at our lecture slides.

Your program should accept the maximum score as a command-line argument. For example, to run it with a max score of 50 points, you would type:

```
$ grade.sh 50
```

If the user does not pass a value for the maximum points, your script should print the following error message and abort:

```
$ ./grade.sh
Usage: ./grade.sh MAXPOINTS
$ chmod 755 grade.sh
$ grade.sh
Usage: grade.sh MAXPOINTS
```

You do not need to check this argument for validity. If an argument is passed, you may assume it is a positive integer greater than 1.

If your script is passed an argument, it will compile each student's source code by running gcc. You may assume all students' source codes compile.

Once they compile, your script will compare the student's output against the expected output file using 'diff -bwi'. If the outputs do not match, consider any line of diff output containing a $<$ or $>$ to count as 1 line of unmatched content. Your program should produce output in the following format:

```
202012345.c has correct output
202012345.c has 3 lines with comments
202012345.c has earned a score of 50 / 50
202012346.c has incorrect output (15 lines do not match)
202012346.c has 2 lines with comments
202012346.c has earned a score of 30 / 50
```

For each unmatched line, you should deduct 1 point from the student's score. In other words, if the diff output contains 8 lines that have < or > in them, the student should lose 8 points. I.e., if the MAXPOINTS is 50, s/he will get 42 points. Note that your grade.sh may deduct two points even if only one line of student's output differs from the output.txt file. This is because diff prints out a line with > from one file and another line with < from the other file. I.e., a single difference may output both > and <.

Your script will also check whether the student has sufficient code comments. Comments are worth 5 points on the assignment. For our purposes, a comment is defined as any line that contains `//`. A student must have 3 or more lines of comments in his/her commands file. A student that has fewer than 3 lines of comments should lose 5 points from his/her assignment score, down to a minimum of 0. Your script should output how many lines of comments are found as shown above.

3 How to Submit

You will need to submit multiple files that you write for this project. Therefore, you should create a sub-directory, for example proj4, in your home, and then run pl_submit command as follows.

```
$ cd /home/2012345678/proj4
$ pl_submit proj4 grade.sh
```

Note that you can submit multiple times. But only the last submission will be graded. Using the following command, you can check whether your file has been correctly submitted.

```
$ pl_check_submission proj4
```

Note that these commands only work in swin, swui, swye, swji machines. If you implemented your codes in your desktop, you must upload the file to these machines before you submit. Otherwise, pl_submit command will fail to read your implementation.

For any questions, please post them in Piazza so that we can share your questions and answers with other students and TAs. Please feel free to raise any issues and post any questions. Also, if you can answer other students' questions, you are welcome to do so. You will get some credits for posting questions and answering other students' questions.