THE UNIVERSITY OF WINNIPEG

APPLIED COMPUTER SCIENCE

ACS-2941-050 - Unix -- Assignment 2

Assignment Information:

Due date: October 31, 2019 — 6:00 PM CT;

- A file containing all answers for this assignment should be submitted by email to Mr. Rajkumar Shaileshkumar Patel (**patel-r19@webmail.uwinnipeg.ca**) with a copy to **l.guidolin@uwinnipeg.ca** using your UW student email;
- If you do not provide an answer to a question, you must add a comment in the preamble of the shell script file indicating which questions were not attempted.
- All scripts must execute on pearl.acs.uwinnipeg.ca, since this assignment will be marked by running the script on it. Students may submit a partially completed assignment and will receive credit for those attempted problems. Students are responsible for maintaining backups of their work.
- This assignment contains 13 questions for a total of 120 possible points. Questions are worth 10 points unless otherwise stated.
- The name of your assignment file must be **`assignment_2_${USER}.sh`**, where **`${USER}`** is your username in pearl.acs.uwinnipeg.ca.

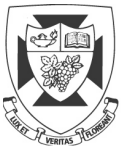
Assignment:

In assignment 1, you submitted your answers in a text file named after your user id as in pearl.acs.uwinnipeg.ca, e.g. **`assignment_1_${USER}.txt`**. In your answer file you identified each question by an **`echo`** statement, as in:

```
echo -ne "Q01:\n"
<replace this text with your answer to question 1>

echo -ne "Q02: <question not attempted>\n"
<add 'question not attempted' to the echo command above if you are not
 providing an answer to this question and leave this line blank.>

echo -ne "Q03:\n"
<replace this text with your answer to question 3>
```

Upon marking your assignment, two additional fields were added to the string in the "**`echo`**" statement indicating 1) how much each question was deemed worth, and 2) a brief explanation as to why the question was not awarded 100% (in a scale of 0 to 1). As an example, considering Q01 as correct, Q02 as not attempted, and Q03 as partially correct (75%), you should expect your marked file to look as follows:

```
echo -ne "Q01:1:\n"
<correct answer provided>

echo -ne "Q02:0:<question not attempted>\n"

echo -ne "Q03:0.75:some reason here\n"
<partially correct answer>
```



THE UNIVERSITY OF WINNIPEG

APPLIED COMPUTER SCIENCE

The format of the string of interest: **`Q##:points:justification\n`**

In this assignment you will create a shell script in bash that will parse these **`echo`** statements, extract strings of interest, calculate the final mark for the assignment, and provide a brief report with the justification for incorrect/partially correct questions.

You can use sample assignment 1 files in /home/lcguid/assignment2 (or any readable and valid assignment file) to test your script. You can compare your script's behaviour against the expected output in the assignment2 supplemental material.

Q01. Your script must start with **`#!/bin/bash`** and **`exit`** returning **`0`** to the shell unless your script identified an error.

Q02. All exceptions, within reason, must be handled by your script and values different than 0 should be returned as exit code(s).

Q03. Use the command **`getopts`** to process parameters passed via command line. Your script should handle three options: **`-h`**, which displays programs usage (see Q05), **`-f file`**, which receives the file name of the assignment 1 file to be processed, and **`-r`** that indicates that a a report of incorrect/partially correct questions should be generated (see Q11).

Q04. Write a function called **`wrong_use`** that outputs the message "**`Try '<script name> -h' for more information.`**" to the standard error. Note that **`<script name>`** should not be hard coded but replaced with the appropriate shell variable that provides the script name.

Q05. Write a function called **`usage`** that displays the following message to the standard output:

```
Usage: <script name> -f file [-r]
    -f file, specify input file
    -r, generate a report
```

Note that **`<script name>`** should be hard coded but replaced with the appropriate shell variable that provides the script name.

Q06. Write a function called **`get_student_name`** that extracts user login name from the file provided as input to the script. This function should receive the variable containing the file name as a parameter and return the user name. Example:

```
# Considering: file=assignment_1_uYxHssnA-l.txt.txt
function call: get_student_name ${file}
returned value: uYxHssnA-l
```

Q07. Write a function called **`parse_assignment_file`** that extracts strings of interest from the **`echo`** commands. The function should receive the file name as input and return all string of interest as output. Example:

```
file: assignment_1_uYxHssnA-l.txt
file content:
      echo -ne "Q01:1:\n"
      ls -la

      echo -ne "Q02:0:question not attempted\n"

      echo -ne "Q03:0.75:did not pipe stderr\n"
      ls -z | grep "Try"

function call: parse_assignment_file ${file}
returned value:
      Q01:1:\n
      Q02:0:question not attempted\n
      Q03:0.75:did not pipe stderr\n
```

Q08. Write a function called **`calc_num_of_points`** that receives the output from the function **`parse_assignment_file`** and calculates the total number of points awarded to the assignment (note that it should be between 0 and 124 points). Suggestion: use the command **bc** for your calculations.

Q09. Write a function called **`print_final_mark`** that receives the number of points awarded to the assignment, as calculated in question Q08, and calculates the final mark as a percentage (0-100%).

Q10. Using functions in questions **Q06** and **Q09** print the final mark to the standard output in the following format:

example output: **`Final mark for student id [uYxHssnA-l] is 84.0%`**

Q11. Write a function called **`make_report`** that receives the output from function in Q07 and prints a report for incorrect or partially correct answers to the standard output. Expected output:

```
Q02 [0] -> question not attempted\n
Q03 [0.75] -> did not pipe stderr\n
```

Q12. (5 points) Your script should return exit code 66 and display a "wrong usage" message if the parameter **`-f file`** was not used (see Q04).

Q13. (5 points) The output of your script should match exactly the output of the example output provided.

-- Unix is user friendly. It's just very selective about who its friends are!