CSC2408 S2 2021 Assignment 2

Git; Shell scripting in *bash*; Markup languages. Due Date: 07 Oct 2021

Objective

The objective of this second assignment is to assess your knowledge of and skills in `bash` scripting and further use of Linux as a platform for software development. It also assesses your foundational understanding of using a remote Git repository in the context of collaboration and teamwork.

Marks

There are **four (4)** questions in Assignment 2, each consisting of multiple parts. There are 60 marks and Assignment 2 comprises 20% of your final mark.

Submission

- Submission is via the USQ Student Git platform on gitea.usq.edu.au. Question 1 contains the steps required to use Git to achieve this requirement.
- You must have at least 5 commits over a period of at least two weeks; see Question 1 for details.
- In your git repository, only the default (master) branch will be marked. If you (optionally) *choose* to use branches, make sure you merge them into `master`.

  The (chronologically) last commit that you pushed to the remote git server prior to the deadline, will be the basis for marking. Later commits pushed to the remote will be ignored.
- For **Question 2**, you will create a `bash` script.
- For **Question 3**, you will create a `bash` script and a `.md` file. These files are mandatory in your repo. The question generates other files and they may be added to the repo if desired.
- For **Question 4**, you will create a LaTeX (`.tex`) document that you will compile into a `.pdf` file. You should add both files to your repository.
- It is expected you maintain the directory structure of the downloaded `.tar.gz`.
- Though you may choose differently, it is *strongly* recommended that you do your work on the Course VM as it contains all the software needed.

Late Submission of Assignments[1]

Students can apply for an extension of time to submit an assignment at any time up to the deadline. Students are advised to make a request for an extension as soon as their need becomes apparent. Delay in making a request involves the risk of losing marks if the request is refused.

Extensions are usually granted only in cases of Compassionate and Compelling Circumstances in accordance with the assessment of Compassionate and Compelling Circumstances Procedure. Generally, extensions will be limited to a maximum of five University Business Days. A Student requiring an extension for a period of time in excess of this should consider applying for a Deferred Assessment as per Section 4.4 of the assessment procedure.

Applications for extensions must be made via the request for extension link. An Assignment submitted after the deadline without an approved extension of time will be penalised. The penalty for late submission without a pre-approved extension is a reduction by 5% of the maximum mark applicable for the assignment, for each University business day or part business day that the assignment is late. An assignment submitted more than ten University business days after the deadline will have a mark of zero recorded for that assignment.

The Examiner may refuse to accept assignments for assessment purposes after marked assignments and/or feedback have been released.

Non-submission of Assignments and Passing Grades[2]

To be assured of receiving a passing grade a student must obtain at least 50% of the total weighted marks available for the course and have satisfied any Secondary Hurdles (if applicable).

Supplementary assessment may be offered where a student has undertaken all of the required summative assessment items and has passed the Primary Hurdle but failed to satisfy the Secondary Hurdle (Supervised), or has satisfied the Secondary Hurdle (Supervised) but failed to achieve a passing Final Grade by 5% or less of the total weighted Marks.

To be awarded a passing grade for a supplementary assessment item (if applicable), a student must achieve at least 50% of the available marks for the supplementary

[1] See http://policy.usq.edu.au/documents.php?id=14749PL for the full assessment regulations.
[2] See the University Assessment Procedure policy document

assessment item as per the Assessment Procedure (point 4.4.2).

The offer of Supplementary Assessment normally will only be made if the Student has undertaken all possible Summative Assessment Items for the Course (i.e. the assignments).

Student Responsibilities

The Assessment Procedure Section 4.2.2, also outlines the following student responsibilities:

- If requested, Students must be capable of providing a copy of Assignments submitted. Copies should be dispatched to the University within 24 hours of receipt of a request being made.
- Students are responsible for submitting the correct Assignment.
- Assignment submissions must contain evidence of student effort to address the requirements of the Assignment. In the absence of evidence of Student effort to address the requirements of the assignment, no Mark will be recorded for that Assessment Item.
- A Student may re-submit an Assignment at any time up to the deadline. A request to re-submit after the deadline is dealt with in accordance with section 4.4 'Deferred, Supplementary and Varied Assessment and Special Consideration' of these procedures.

Academic Integrity

Academic misconduct is unacceptable and includes plagiarism, collusion and cheating:

- *plagiarism*: involves the use of another person's work without full and clear referencing and acknowledgement;
- *cheating*: involves presenting another person's work as your own — this includes *contract cheating*.
- *collusion*: is a specific type of cheating, that occurs when two or more students fail to abide by directions from the examiner regarding the permitted level of collaboration on an assessment.

All are seen by the University as acts of misconduct for which you can be penalised. For further details go to the Library's site on Plagiarism.

Question 1 (4 Marks)

Topic: Git
Submission: Git repository pushed to the remote.

First Read the CSC2408 USQ Gitea instructions carefully, and then point your browser to gitea.usq.edu.au, log in using your USQ student credentials, and use the Gitea web interface to create a new (initialised) git repository named

`CSC2408-S2-21-A2-myname`

where `myname` is your family name followed by your given name (e.g. '`smithjohn`').

Secondly, use the Gitea web interface to invite user '`dekeyser`' as a collaborator for your new repository. Again see the instructions linked above for details.

Third, in your Debian Linux environment, using the command line, clone the repository. Use the `git clone HTTPS` URL that is provided in the Gitea Platform's web interface for your new repo. In your Linux environment, on the command line, you will be prompted for your USQ student credentials when running `git clone`.

Fourth, use the web browser in your Linux environment to download the `Assignment2.tar.gz` file from the CSC2408 StudyDesk assessment page. Untar and unzip the archive into the directory that you cloned the repo in. If you now do a `git status` in your repo working directory, you'll see that all the files you unpacked are unstaged.

Fifth, add the files unpacked from the archive into your git repo, using the `git add` command. If you now do a `git status` you'll see that the files are staged. Now commit the changes, and use commit message "*Added files from the Assignment2 archive*". If you now do a `git log` or `tig`, you should see the first commit in your repo.

Sixth, for **two (2)** of the 4 marks available for this first question, push the contents of your local repo to the remote Gitea server, by using the `git push` command on the command line. You will again be prompted for your USQ student credentials.
As an optional step you can go back to the gitea.usq.edu.au browser GUI interface and double-check that your push was successful.

Finally, you are now ready to start work on Questions 2, 3, and 4. Make sure you periodically `commit` as well as `push` your changes. You must have **at least five (5) commits over a period of at least 2 weeks**. This is required to gain **two (2)** of the 4 marks available for this first question.

As mentioned on Page 1 only the last commit you pushed prior to the deadline, involving the master branch, will be used to mark Questions 2 to 4.

Question 2 (25 Marks)

Topic: `bash` scripting; case, reading interactively.

Submission: You will create a `bash` script called `userid_q2.sh` where *userid* is your student number. This file must be in the `Q2` directory of your repository. More files can be included, but these are the minimum.

Your task is create a `bash` shell script that removes comments and blank lines from another shell script, leaving only the commands.

- You are provided an example shell script `samplebash.sh` on which your own script can be run as a test.
- Remember comments start with a `#` (hash) and can be at the start of a line or the `#` can be on a line after a command.
- Assume that if a `#` is used ***in*** a command (an `echo` or `if` statement for example) that there will be no comment also on that line. See the example `samplebash.sh` for clarification.
- Assume that the *shebang* (`#!/bin/bash`) is the first line of the script and never contains a comment.

Your tasks for this assignment:

1. Write a `bash` script called `userid_q2.sh` where `userid` is your student ID. The script takes two arguments, an input file (e.g. `samplebash.sh`) and an output file.
    - Verify that two arguments are provided to the script, otherwise exit (2 marks)
    - Verify that the input file exists and can be read by the script, otherwise exit (2 marks).
    - If the output file *aleady exists, and is writable by the user*, the script should ask the user if it is OK to overwrite the output file. The user can respond with `y`, `Y`, `n`, `N`. If any other response is given, the script should ask again until a valid response is received. (6 marks)
    - As part of the previous requirement, you should use a `case` statement. (2 marks for using `case`)
    - The script should remove all comments from the input script and write the script without comments to the output file. The script should not affect commands that contain the `#`. (8 marks)
    - The script should remove all blank lines from the input script. (2 marks)
    - The script should make the resulting output script executable by the owner. (1 mark)

2. Your `userid_q2.sh` script should contain comments which explain the code for the reader. (2 marks)

Notes:

- Commands used in the script must be standard Linux utilities as described in the course Workshops.
- Ensure your script uses relative (not absolute) paths whenever possible; the marker will be cloning your repository (and running your script) on their own computer in a directory hierarchy that is different from yours.

END OF QUESTION 2

.

Question 3 (25 Marks)

Topic: `bash` scripting and Markdown; handling files

Submission: You will create a `bash` script called `userid_q3.sh` where *userid* is your student number and a (Github-flavoured) markdown file called `overviewTemplate.md`. These two files *must* be in the `Q3` directory of your repository; other files may be included.

You are an electric car sales person. Your task is to create a `bash` shell script that reads in car test marks and creates an overview PDF file that includes the marks, comments, car IDs and car description. Your script will read in a template `.md` document from which to create the output PDFs (by substituting placeholders with values).

- You are provided two comma-separated files; `Cars.txt` which contains car IDs and car descriptions, and `Marks.txt` which has a car ID, marks for each of 3 tests, and some comments.
- You do not need to error-check the contents of the files. For example, you can assume that the car IDs are present correctly in each file and the marks are correctly integers 0-10.
- All car IDs are assumed to start with an `ev` prefix (e.g. ev0217).

Your tasks for this assignment:

1. Create a `.md` document called overviewTemplate.md. Your template `.md` document should compile using `pandoc` and look as similar as possible to the example `overviewTemplate.pdf` provided. It may compile with *warnings* but the compile should not stop with *errors*. (6 marks)
2. Write a `bash` script called `userid_q3.sh` where `userid` is your student ID. The script takes no arguments, and it must perform the following tasks:
   - Verify that the input files `Cars.txt` and `Marks.txt` exist and can be read by the script, otherwise exit. (1 marks)
   - Verify that `Marks.txt` is not empty, otherwise exit (1 marks)
   - Verify that `overviewTemplate.md` document exists, otherwise exit (1 marks)
   - Verify that the output directory OUTPUT exists, otherwise exit (1 marks)
   - For each car, the script must produce a `.md` document which replaces the placeholder information (the car's marks for each test, total marks, comments, car ID and description) in the `overviewTemplate.md` file. This file should be called `carid.md` where `carid` is the current car's id number. (6 marks)

- Compile the `.md` documents using `pandoc` and the resulting `.pdf` file should be placed in the `OUTPUT` directory. (3 marks)
- Clean up any intermediate files (e.g. `.aux`, etc), and remove the created `carid.md` document. (2 marks)
- At each step, inform the user of what is happening. All other output and errors from other commands should be suppressed. (2 marks)

3. Your script should contain comments which explain the code for the reader. (2 marks)

Notes:

- You should make use of `sed` in this question. You may use piped `sed` commands or a `sed` script if you prefer.
- Commands used in the script must be standard Linux utilities as described in the course Workshops.
- Ensure your script uses relative (not absolute) paths whenever possible; the marker will be cloning your repository (and running your script) on their own computer in a directory hierarchy that is different from yours.

END OF QUESTION 3

.

Question 4 (6 Marks)

Topic: LaTeX document typesetting

Submission: You will create two files `userid_q4.tex` and `userid_q4.pdf` where *userid* is your student number. These files must be in the `Q4` directory of your repository. Make sure none of the LaTeX intermediate files (such as `.aux`) are included in the repo.

You are a tutor in CSC1117 *Intro to Computing*. Your task is to prepare a document in LaTeX introducing yourself and mentioning some of the things you are going to be teaching in the coming semester.

Your tasks for this assignment:

1. Create a `.tex` document called `userid_q4.tex` where `userid` is your student ID. You must compile this with `pdflatex`. It may compile with *warnings* but the compile should not stop with *errors*. (1 marks)

2. Have a title area which has your name, the course name and ID and semester. You do not necessarily need to use the `\maketitle` command for this task; simply have the text laid out nicely. (2 marks)

3. Create a **section** containing a small biography of yourself, including your education, research interests and other teaching for the coming semester. (Note: *This information can be fictional; please do not reveal real personal information if you do not wish to!*) (1 mark)

4. Create a second section containing some information about the course this semester. There should then be two **subsections**:

   - One subsection contains some maths. This could be anything you like, such as the famous $e = mc^2$. (1 mark)
   - The other subsection contains a table. Its content should be as in the `overviewTemplate.md` as specified in Question 3, though with improved styling (e.g. cell alignment). (1 mark)

Notes:

- The exact layout and things like section headings are up to you. You may use various LaTeX packages but it would be preferred that you use packages which are installed on the Course VM or which are installed using the commands provided for installing TeXLive in the Pracs. If you are using packages which you think may be unusual, please place a comment in your TeX document.

END OF QUESTION 4

End of Assignment 2

You should have several files in your repo as indicated.
Ensure that any work in branches is merged into the master branch.
Ensure you periodically push your commits to the remote (USQ Gitea), and your last one is pushed prior to the deadline.

For your peace of mind, after you `commit` and `push` to USQ Gitea, you can use your browser to visit the Gitea web interface and verify that all files are there as you expected.

This assignment specification was typeset using LaTeX