CS108-Bash Project Report

Udit Sarkar

April 28, 2024

Contents

1 Objective

This project is a bash script that manages and interprets CSV files containing student exam data. The script provides various commands to combine, upload, calculate totals, and version control the CSV files.

2 Basic Logic Of The Code

The submission.sh file contains many functions such as combine,total etc. The csv files to be interpreted should be in the same directory as the submission.sh. When the user will run the bash file, a menu will appear in the terminal, user can select the desired function to be implemented.

All the functions the user will implement, the changes will reflect on the main.csv file present on the same directory.

It also includes functions for generating visualisations, generating statistics etc. So the corresponding png file,txt file etc. will be added in the same directory.Also there are line.py and bar.py that are python codes that generate line and bar graphs.

3 Working Of the Code

The explanation of the all functions included in the submission.sh are as follows:

3.1 combine()

This function will combine all the csv files(except main.csv) present in the same directory creating new columns with column name as the basename of the csv file.To pop main.csv I have used reference : [1]

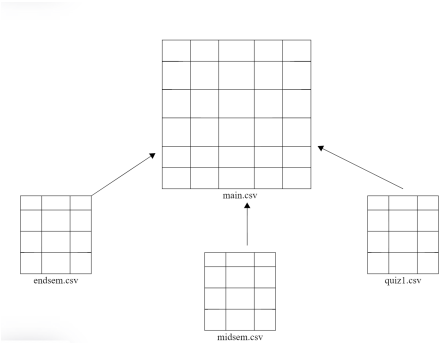

Figure 1: combine diagram

Usage: bash submission.sh combine

3.2 upload()

This function will take path of the file from the user and upload that file to the directory same as submussion.sh . It may also take files that are not csv.

Usage: bash submission.sh upload <file-path>

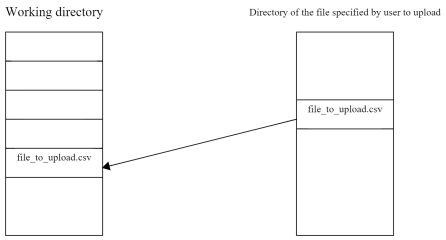

Figure 2: upload diagram

3.3 total()

This function will append the sum of marks of individual students in a new column 'total' in main.csv

Usage: bash submission.sh total

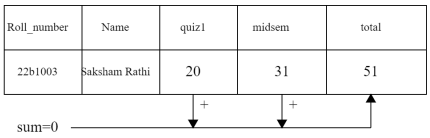

Figure 3: total diagram

3.4 git_init()

This function will initialise a remote repository taking the path of the repo as user input. The path of the remote repo will be redirected to .git_remote file

Usage: bash submission.sh git_init <remote-repo-path>

3.5 git_commit()

This function will copy all the files from the working directory to the remote repo inside the directory named by creating a hash value.

Usage: bash submission.sh git_commit

3.6 git_checkout()

This function will copy all the files from the directory inside the remote directory (initialised in git_init) , after taking user input as hash value or commit message or the prefix of the hash value.

Usage: bash submission.sh checkout <hash-value/prefix/commit-message>

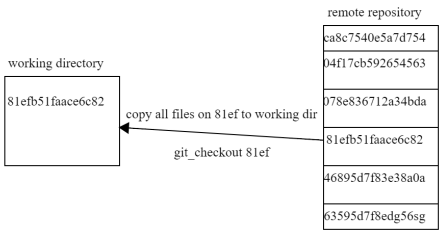

Figure 4: checkout diagram

3.7 update()

This funtion will take the roll number,name and the name of the exam user want to update as user input and update the main.csv and the corresponding csv file.**Note:** Enter the roll number having lower alphabets.
Usage: bash submission.sh update

3.8 update_all_marks()

This function will be used when the user wants to update all the marks of a particular student taking roll number and name of the student.This will automatically update the corresponding csv files as well. References used : [2]

Usage: bash submission.sh update_all_marks

3.9 git_commit_dp

This function will copy all the files from the previous commit directory to the new commit directory, then create a .diff file in the new commit directory and

patch the modified files.If it is the first commit, then it will directly copy all the files from the working directory to the new commit directory.This will also display the files modified.[3] [4] ; Rerference for creating hash value: [5] ; Referernce used for error handling in patch: [6]

Usage: bash submission.sh git_commit_dp

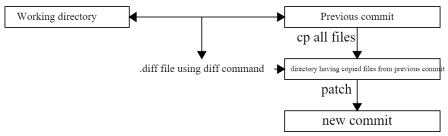

Figure 5: commit diagram

3.10 list_exams()

This function will list all the exams and total number of exams held till now.

Usage: bash submission.sh list_exams

3.11 replace_absent()

This function will replace all the 'a' ie absent to '0'.

Usage: bash submission.sh replace_absent

3.12 fix_weightage()

This function will help the user to fix the weightage of different exams redirecting to the weightage.txt that is to be used in the further calculation of grade. References used: [7]

Usage: bash submission.sh fix_weightage

3.13 generate_relative_grades()

This function will append a new column 'grade' calculated relatively (according to percentile) according to weightage.txt or from user input to the main.csv.User can only use this after running the total function.

Usage: bash submission.sh generate_relative_grades

```
Roll_Number,Name,endsem,midsem,quiz1,total,grade
22b009,Mayank Kumar,a,07,7,14,CC
22b1003,Saksham Rathi,1,22,1,24,BB
23b0069,Sunny,9,15,4,28,AA
23b0108,Ramesh,a,24,a,24,BB
```

Figure 6: grades

3.14 git_diff()

This function will show the difference between the files present in the directories as to the user input as hash value or commit message or the prefix of the hash valueinside the remote repo.

Usage: bash submission.sh git_diff

3.15 statistics()

This function will generate the statistics ie mean,median and standard deviation of the 'total' column of main.csv file and print these to the termimal and also redirect to the stats.txt file of the working directory. References used : [8]

Usage: bash submission.sh statistics

```
Mean: 22.5 ; Median: 24 ; Standard Deviation: 5.17204
```

Figure 7: stats

3.16 visualize()

This function will genarate line and bar graphs and save thees to line.png and bar.png respectively of the data present in the main.csv.To check for dependecies: [9] ; other references : [10]

Usage: bash submission.sh visualize

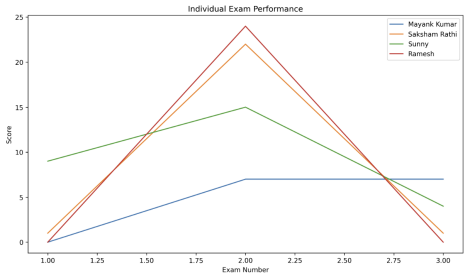

Figure 8: Line graph

3.17 menu()

This function will print a menu in the terminal which make it easier for the user this bash gradder file.

Usage: bash submission.sh menu

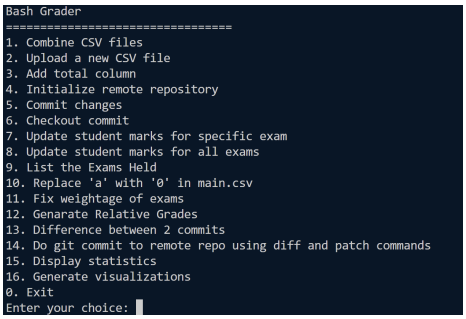

Figure 9: menu

4 Utilities

- Python : For visualisation
- sed,awk : For data manipulation from different csv files

5 Customisations

The customisations include:

- list_exams 3.10
- fix_weightage 3.12
- genarate_relative_grade 3.13
- git_diff 3.14
- statistics 3.15
- visualize 3.16
- replace_absent 3.11
- git_commit_dp 3.9

References

[1] In: (). URL: https://stackoverflow.com/questions/33297857/how-to-check-dependency-in-bash-script.

[2] In: (). URL: https://stackoverflow.com/questions/11531693/patch-only-garbage-was-found-in-the-patch-input.

[3] In: (). URL: https://www.scaler.com/topics/linux-patch/#.

[4] In: (). URL: https://stackoverflow.com/questions/9980186/how-to-create-a-patch-for-a-whole-directory-to-update-it.

[5] In: (). URL: https://ioflood.com/blog/bash-random-number/.

[6] In: (). URL: https://stackoverflow.com/questions/11531693/patch-only-garbage-was-found-in-the-patch-input.

[7] In: (). URL: https://stackoverflow.com/questions/53093449/convert-text-file-into-a-comma-delimited-string.

[8] In: (). URL: https://unix.stackexchange.com/questions/13731/is-there-a-way-to-get-the-min-max-median-and-average-of-a-list-of-numbers-in.

[9] In: (). URL: https://stackoverflow.com/questions/33297857/how-to-check-dependency-in-bash-script.

[10] In: (). URL: https://www.geeksforgeeks.org/data-visualization-with-python/.