CS-108 (Bash Project)

Ashish Kumar (23B1028)

April 2024

Contents

Chapter 1

Introduction

This project is indent to create a csv file manager and interpreter. Let's understand the work of this project by an example. Let's say you have so many files (csv) consists of marks of students in the exams in this format :- "Roll number,Name,marks" and you want to merge these file into one main.csv file in which you have to store marks of all students from the different file to main.csv. This is what the project is all about. This project has 2 sections 1.Auto Grader, 2.Version Control system. Let's understand each one by one.

Chapter 2

Auto Grader

This section has 4 necessary function and some customize function Which has been written in submission.sh , function_auto_grader.sh file in the working Directory. And for graphical visualisation 1 plotting.py file to plot the graph for stats of student.

2.1 Combine

combine : In this function I have created a main.csv file which contains the data for all the exams so far. So, on running this command, The bash script will iterate over all the csv files in that directory and will combine to create main.csv. So, let's suppose you have three files: quiz1.csv, quiz2.csv, endsem.csv. Then main.csv will have the following columns: "Roll_Number,Name,quiz1,quiz2,endsem" (Assuming the roll numbers to be case-insensitive). So, we add a column corresponding to every file in the main.csv file and add corresponding marks in the corresponding column. Moreover, If a particular student is not present in that exam than it will mark it as "a" (absent). (Assuming, Roll numbers are unique).

- Command to run : **bash submission.sh combine**

2.2 Upload

upload : You can also upload a file from the different file location to the script directory.

- Command to run : **bash submission.sh upload /path/to/that/file**

2.3 Total

total : On running the command "total", your script should create a new column (with heading "total") in main.csv, which stores the total of every student. (with

absent being treated as 0). After adding total then also you can run combine command to combine more files.

- Command to run : **bash submission.sh total**

2.4 Update

update: What if the corresponding TA wishes to increase marks after cribs? For That I have develop an update function which takes input as "Roll number, exam name, new marks" for updating the marks of corresponding student in the main.csv and the corresponding "exam name.csv" file .

- Command to run : **bash submission.sh update**
- This will automatically ask you for roll number exam and the updated marks.

2.5 Stats

when you write this command after combining data to main.csv then it will create the statistics of all student in each exam like mean, median and standard deviation in stats.csv file in that directory.

- Command to run : **bash submission.sh stats**

2.6 Plotting

This part of the submission.sh will run python script (plotting.py) which uses main.csv and stats.csv to show the graphical representation for the stats of students. It has two sections

2.6.1 Each student marks

This section of the python script will take the roll number of student as input and give the double bar graph of the student marks in each subject and the highest mark obtained by student in each subject. (This will use main.csv to for output).

- Command to run : **bash submission.sh plot**

2.6.2 All student performance

This section of python will take exam name as input and give the bar graph. For Mean, Median and Standarad deviation of all the student in that exam. (This part of the script is using stats.csv for output).

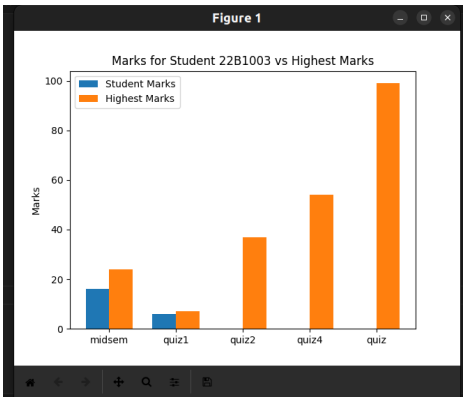

Figure 2.1: Example image of plot command for each student marks.

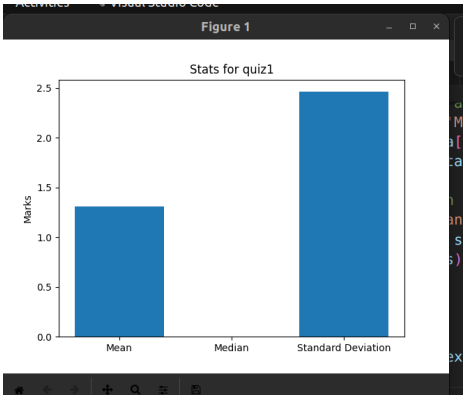

Figure 2.2: Example image of plot command for all student performance.

- **After giving the roll number and closing the graph for the first part the script will ask you for the exam name (on terminal).**

2.7 Rank of students

This part of submission.sh will print out the rank of student Name,Roll_Number,Total,Rank .

- Command to run : **bash submission.sh rank**
- **You can't run update command after stats, rank or plot command.**
- **Plot command is only executed after stats.csv and main.csv is created.**

Chapter 3

Version Control System (Git)

This section of the bash script is to replicate git to store the version of the files. This section contain 3 main command and some customisation which is written in submission.sh and function_git.sh .

3.1 git_init

git_init: This command initializes the remote directory. So, suppose you type, "bash submission.sh git_init /Desktop/bc", then your script will mark the folder "bc" on Desktop as the remote repository. (It will create such a folder, if it does not exist already!)

- Command to run : **bash submission.sh git_init /path/to/the/repository**

3.2 git_commit

git_commit : This command copies the current version of all the files to that remote directory. If git_init has not been executed already, then this will print an error message and stop executing the script. This command can be executed as follows: **bash submission.sh git_commit -m "commit-message"** . Every commit will generate a random number (of 16 digits) (also known as hash value). This number will act as an id of that commit. All these hash values and the corresponding commit messages will be stored in a .git_log file present in the remote directory. Also, It prints the names of all the files which were modified after we did the last commit. (except for the first commit).

3.3 git_checkout

git_checkout: This command reverts our current directory back to the commit message we specify. This command can be executed as **bash submission.sh git_checkout "commit-message"** or **bash submission.sh git_checkout "hash_value"**. Also, No need of giving the complete hash value, even if some prefix of that value is provided, the script will checkout to that commit. (It will return a conflict if two different hash values start with the same prefix or same name message for two commits).

- Note : If any type of confict occur then use git_log to get the unique hash values of the commits you have made so far . (very useful)

3.4 git_log

git_log : This command will print all the "commit messages" and the "hash value" for each commit that were made by the user in that remote repository.

- Command to run : **bash submission.sh git_log**.

Chapter 4

References

- **For Python script** :- go (https://matplotlib.org/) or Click here and Kameswari Chebrolu. CS 108 Course Slide.
- **For Bash script** :- go (https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/) or Click here and Kameswari Chebrolu. CS 108 Course Slide.
- **For sed and Awk** :- ChatGpt and Kameswari Chebrolu. CS 108 Course Slide.