# CS108 Project-Bash Grader

Shiv Narang

April 28, 2024

# Contents

# 1 Objective

The bash file submission.sh implemt a csv file manager using bash scripting and smaller version of how git controls version management.Another files of python has some added customizations

# 2 Introduction to Bash Grader

Bash Grader is a csv file manager and interpreter. A csv file is comma sepearted file.Also it has smaller version of git implemented using bash scripting and data analysis of students is being done using python and modules

# 3 Applications

Some of the Applications of bash grader is

- Data Analysis in schools and colleges This can used for fast and efficient data analysis of students in the college and also grading can be done efficiently using this
- Automated Testing This can be implemented when we have large set of data and we have to automate tests on that data.

# 4 Basic Idea of the code

Basic idea of the code is if you have multiple csv files in your current directory it can combine all the csv files into main file and also can add total column in that file which is sum over all csv files.We can also upload new files and can update marks of students in different csv files.
Version controlling is also being implemented.You can initialize remote repository and commit and checkout your versions in that repository. And at the last some customizations is also added using python and modules

# 5 Working of the code

The bash language is used for performing different tasks.The working of all commands in bash is being briefly explained using comments also use of variable name is also descriptive to make it easier to understand.There is a main file submission.sh which control all the functionalities i have implemented

## 6 Utilities Used

I have used basic and advanced UNIX commands along with sed and awk in my bash scripts and for customisation i have used python modules numpy,pandas for loading data,tkinter for ui design,matplotlib pyplot sandbox for graphs,sys module for capturing command line argument

## 7 Break down of files

- submission.sh It is a master file that controls all commands and functionalities implement its usage is explained later in report
- combine.sh As the name suggest this file is used for combining all csv files and it is called by submission.sh
- total.awk This awk file is used to add a total column in my main.csv file .This is also called by submission.sh
- update.sh This bash file is used for updating cribs of the student it dynamically changes marks in that particular csv file and main.csv file
- git.sh This file is for implementing all the basic and advanced git commands that have been used and invokes by submission.sh
- advancedgit.sh This file is for implementing all the advance version of git using staging area
- analysis.py This file is for analysing data in user friendly way , ui design created using tkinter module of python also graphs and statistics is implemented in this file
- overallanalysis.py This file is for overanalysis of the students in all exams and tkinter and plots is implemented in this file

## 8 Usage of basic commands and customised command

The usage of the files is simple. Open a terminal on your computer. Change the directory to the location where the bash file is present. Ensure that you have the following libraries installed:numpy,matplotlib,tkinter,pandas,math,sys,sns. Now run the following command on the terminal.
**"bash submission.sh command any other extra arguments(if needed)"**

- **bash submission.sh combine** this will combine all the csv files and combine them to main.csv file

- **bash submission.sh upload address_of_file** This will upload that file into my current directory

- **bash submission.sh total** this will add total column in main.csv file

- **bash submission.sh update** This will give option to update marks in different exams and will update the marks of student given by user through inpu

- **bash submission.sh git_init repo_path** This will generate a remote repository in the path given by user as command line arguments

- **bash submission.sh git_commit -m ”commit message”** This will create a commit for that commit message in remote repository

- **bash submission.sh git_checkout -m ”commit message”or git_checkout hash_value**

- **submission.sh analysis Roll_Number** This will open a ui which will prompt user to chose exam he want analysis of than it will show graph and stats

- **submission.sh overallanalysis.py** To analyses all exams

- **bash submission.sh git__init repo_path For all the advanced git functions i have used __ to avoid confusion** This is to initialise repository for advanced git commands that are implemented

- **bash submission.sh git__commit -m message** This is to do commit in advanced git

- **bash submission.sh git__add file1 file2 or .** This is to add files in staging area

- **bash submission.sh git__status** This is to display the untracked,modified and files in staging area

- **bash submission.sh git__log** This is to show commit history

- **bash submission.sh git__show : filename or git__show 1-16digitsofcommitid filename** This is to show the content of file in staging area or in commit history

## 9 Customization

- **Optimised combine command**
  If there is no file which is being uploaded or there is no new csv file than combine commmand will not run again and it will display that you already have upto date version of main.csv file and if main.csv have total column in it and i have uploaded new file than it will have total column entact

- **Optimised total command**
  If there is already total column present i will not recalculate total if total command iis given again

- **Optimised update command**
  User will prompt to write rollno and student.If valid student than it will give options for exams to update Then it will update marks of student in that exam also the marks would be changed in main.csv file and if it has total column that it will change total column also

- **Optimised basic git commands**

  - **git_init** Added author name and email in commit history
  - **git_commit** Implemented **Diff and Patch command** for storing commits so my logic is that for first commit i will copy all csv files in that commit folder which is in remoterepo and for later commits i will compare the files in my current directory with first commit and if the file differs i will store patch file corresponding to it.It also displays files which are modified from last commit
    Also i have included the case if new files added or delted
    If new file is added than i will store empty file corresponding to it in first commit and then take diff from that file it also displays which file is added
    If any file is deleted than i will take diff from that file in first commit and empty file so that while checking out patch file empty the files it also displays which file is deleted
  - **git_checkout** If there is conflicting commitid than i am giving options to user to select from commitid which he wants to checkout
    Optimised checkout command so first i will remove all csv files from my current directory than copy all csv files from first commit and then use patch command to modify all files whose patch file is being stored in the commit that i am checking out

- Created a **UI** display for user using tkinter module of python It gives options for user to select exam which he wants to analyse that on clicking on analyse button it displays another window which is used to display statistics and graphs of that exam

- Created graphs using matplotlib for effective data analysis Used matplotlib pyplot to draw **HISTOGRAM** and **BOXPLOT** for the students who were present in particular exam promted by user.Integrated these graphs inside tkinter display and tried to make graphs more beautiful and detailed labelings.In boxplot i have plotted Mean,Median and User marks also

- **Statistics of the User**

  - Use pandas module of python to load data of csv files in python and then calculated Mean,Median,Standard Deviation,25th percentile,75th percentile,Highest,Lowest marks usng numpy in built functions.

  - Also i have displayed the number of absent and present students present in that exam using list methods.
  - Displayed Marks of user by filtering data of student from dataframe and then storing details in dictionary
  - Also displayed rank of student in that exam using sorted option of dataframe
  - Also provide the top 5 students of that exam using head option of dataframe

- **UI display of overall exams** Also create ui for the overall analysis of students using tkinter module and plot the seaborn boxplot for overall all exams and also providing overall details of any student inn that file
- **Implemented Advanced git system** I have implemented concept of staging area in my advancedgit.sh file which first adds files in staging area then later on commit.Modified my gitinit gitcommit and gitcheckout functions in accordance with staging area
- **Added git__add command** Now in this git version you cannot directly commit the files first you have to add them in staging area and then commit from there , also added . option with it so that all modified files in my directory can be added directly
- **Added git__status in advanced git** Added git status command to manage untracked files modified files or files present in staging area beautifully.It tells the file you have added in staging and is ready to commit also any modified files which are not added in staging area and untracked files
- **Added git__log** To display all the commit history of the file
- **Added git__show** To display content of file present in that specific commit history

## 10 Basic logics of all commands implemented by me

- **combine** For this i will start iterating from csv files present in my current directory and then if i found new student i will echo this student along with strings of ”a” before its record and if student is present already in main.csv file i will substitute marks to end of that student field.And at last i will see if there is any student who is in main.csv but not in that particular file that means he was absent so i will substitute a in his mark

- **total** simply calculated total using awk script .
- **upload** simple just check file is csv and exist or not and then copy
- **update** For updating i am prompting user to give input wether he want to change marks in particular exam or not.If yes then i will use awk to change that marks in particular file and to change file in main.csv i will first capture the column number of that exam in main.csv and after that i will use awk to change that value and adjust the total column if total column is present
- **git_commit** My logic is that for first commit i will store all the csv files in that commit folder but for later commits i will take first commit as reference and generate patch file if there is a difference in file version of first commit and the current working directory.
  I have also included the case of upload and delete file
  If i add a file in any stage of commit than i will create empty file of that name in my first commit and take diff of empty file with the current version of newly added file and store it in that commit
  For delete case I will iterate through files of first commit and if file is not present in my current dir that means it is deleted and so i will create empty file whose name is of deleted file and take diff of file in first commit and empty file of deleted file(I will later delete it) and store that patch file in current commit directory.
- **git_checkout** As in problem statement we want that during checkout our directory should be in same condition as checkout commit so i will remove all csv file from my current directory .Then i will copy all files from first commit to that folder except the empty files for the empty file i will check if there is patch file corresponding to that file present in commit directory(commit which is getting checkout) if patch file not present that means this file was added in later on commits and then i will use patch files present in my checked commit directory to modify files in my working directory.If after patching i found that file is empty that means this file was deleted in that commit
- **analysis.py and overall analysis.py** I have briefly explain about these file in the comments and also in customization section it is creating ui and showing satistics and graphs of students
- **advancedgit.sh** In this file i have implemented idea of staging area so first you have to add your file in staging area for commit. You can see the progress of files and untracked modified files usin git__status whose function is to tell which file is modified and which are ready for commit.I have not implemented concept of patch in this because it was getting extremely hard to cater all the cases which are possible.Durring commit and

checkout i will simply copy the csvs files but gitadd and gitstatus functioning was tough to implement and took long time.For detailed functioning i have done heavy commenting in the code

## References

[1] Kameswari Chebrolu.CS 108 Course Slides Python Matplotlib Bash Sed Awk Git 2024.

[2] Tkinter https://www.geeksforgeeks.org/python-gui-tkinter/

[3] Chatgpt

[4] Google.com