BASH GRADER

23B0983

Pushpa Kumar

April 28, 2024

Contents

1 Objective

To create a bash program that manages and interprets CSV files easily with many other features.

2 Introduction

The bash file **submission.sh** helps to manage CSV files and stores data of students in **main.csv**.Python file **analyse.py** analyses marks of students by graphs, mean, median ... etc and it even implements basic git.

We have marks of students in different files from which we need to analyse.

A wide range of options

- Combine
- Upload
- Total
- Search
- Update
- GIT
  1. git_init
  2. git_commit
  3. git_checkout
  4. git_log
- Analyse

3 Combine

usage : bash submission.sh combine

3.1 Basic Idea Of Code

Combine basically combines all students marks in different exams and displays in **main.csv** file.

Logic

It performs the following steps:

- Creates a header in the output file containing "Roll Number", "Name", and exam names from each input file.
- Appends the first two columns (Roll Number and Name) from each exam file to the output file.
- Removes duplicate entries based on the combination of Roll Number.
  - Just use 'sort -u'

- Stores scores of each student in the output file by iterating through each exam file:
  1. If a student's Roll Number is found in the exam file, respective exam score (3rd column) is appended to their line in the **main.csv**.
  2. If a student's Roll Number is not found in an exam file, "a" (Not Available) is appended instead.

Handles different scenarios for **main.csv :**

- If **main.csv** doesn't exist, it creates a new file with the appropriate header and add marks.
- If **main.csv** exists but doesn't contain a "Total" column,same as above.
- If **main.csv** exists and contains a "Total" column, it even executes ('bash submission.sh total') as total must be retained after combining.

4 Upload

usage : bash submission.sh upload <path to file>

This **upload** command allows users to upload a file to their current directory from the command line by providing the file path as the second argument when running the script.

5 Total

usage : bash submission.sh total

The **Total** functionality within the script recalculates the total marks for all students in the **main.csv** file. It achieves this by:

- Checking for Existing 'Total' Column in **main.csv**:
  1. If "Total" is already present
     - The script assumes the totals have been calculated and avoids redundant processing.
  2. If "Total" is missing:
     - It calculates Total through external script **total.awk** by iterating through rows
     - First output is stored in **temp** file and then it is moved to **main.csv**

6 Update

usage : bash submission.sh update

This **update** allows a Teaching Assistant (TA) to update marks for a specific student in a chosen exam file and potentially add a new student to the exam. It then reflects these changes in the **main.csv** file.
First it updates marks in each exam file later in **main.csv**

1. Prompts for Roll Number and Exam Name
   - Reads the roll number and exam name to search from the terminal and stores it.
   - All available exams are displyed for clarity

```
(base) pushpakumar@pushpa-kumar py % bash sub.sh update
Enter roll_no: 23b0987
Exams available: midsem, quiz1, test,
```

Figure 1: Displaying all exams

2. Searching for Exam Name:
   - It verifies if the exam file exists using find
     (a) If it is present proceeds further
     (b) If it is not present prints an error message indicating the entered exam name is invalid
3. Updating Marks: If exam is present

   If student roll number is already present in that exam
   - The user enters the updated marks.
   - sed is used to modify the specific line in the exam file by replacing the existing marks with the updated marks.
   - The modified content is written to a temporary file and then renamed to replace the original exam file.
4. Adding New Student: If exam is present
   - If the student's record isn't found in the exam file, the user(TA) is offered to add a new student.
   - It checks if the roll number exists in the main CSV (**main.csv**) using **search_rollno** function.
     (a) If found, the student's name is extracted from the search result.
     (b) User enters the obtained/updated marks.
     (c) A new line with student information and marks is appended to the exam file.

```
(base) pushpakumar@pushpa-kumar py % bash sub.sh update
Enter roll_no: 23b0983
Exams available: midsem, quiz1, test,
Exam name to update: quiz1
enter updated marks: 23
```

Figure 2: Updating marks

   - If the roll number doesn't exist in main.csv, the user is prompted to enter the student's name.
     (a) User enters the obtained marks and Name of the student.
     (b) A new line with student information and marks is appended to the exam file.
   - Combining Updates:
     (a) Above code updates marks of students in that particular exam files
     (b) To reflect in **main.csv** run (**bash submission.sh combine**).

```
(base) pushpakumar@pushpa-kumar py % bash sub.sh update
Enter roll_no: 23b0983
Exams available: midsem, quiz1, test,
Exam name to update: midsem
would you like to add new student in midsem (y/n) :y
enter marks obtained  :45
```

Figure 3: Adding new student

7 GIT

7.1 git_init

usage : bash submission.sh git_init <path>

Initialises remote repository

- Checks for two arguments
- Stores path in .dir file
- git_init status is stored in .git_status.
- Creates directory if not present based on the path given

7.2 git_commit

usage : bash submission.sh git_commit -m "message"

Commits pwd and stores it in remote repository

- Checks if git_init has been run beforehand,if not exits.
- For first time it asks user info and stores it in .git_user.

```
(base) pushpakumar@pushpa-kumar submission % bash submission.sh git_commit -m "first"
enter your name :pushpa
enter you email :pushpakumar222@gmail.com
commited
```

Figure 4: Collecting user information

- Generates 16 digit hash
- Commit message and hash is stored in .git_log, commit time is stored in .git_date.
- Checks no of commits
  - If no of commits is 1 it directly copies pwd to repositry.
  - If no of commits is greater than 1, it uses patch to modify changes (here i wrote separate functions to remove those files deleted in pwd and to add files which are added in pwd as patch by default doesnt consider those)
- Displaying difference between previous and present commit
  - Displays files added/deleted and modified separately.

```
(base) pushpakumar@pushpa-kumar submission % bash submission.sh git_commit -m "fifth"
patching file main.csv
patching file test.csv
```

Figure 5: Patching files

```
-------------------------------------------------------
Files added or deleted :
hello.txt
hello1.txt

Files that are modified :
main.csv
test.csv
-------------------------------------------------------
```

Figure 6: Diff between present and previous commit

7.3 git_checkout

usage : bash submission.sh git_checkout -m "message"
or
usage : bash submission.sh git_checkout hash

Checkout to previous versions..

1. No of arguments is two,
   - Part of hash is sufficient if there are more than 2 hashes it throws error
2. No of arguments is three,
   - Searches based on message and checks out.

7.3.1 Checkout MASTER

usage : bash submission.sh git_checkout master

- Checksout to latest commit.

8 git_log

usage : bash submission.sh git_log

Shows commit history

```
    44d77b287e1a4aa9
commited by  pushpa <pushpakumar222@gmail.com>
Date: Sat Apr 27 22:45:52 IST 2024
    second
```

Figure 7: example of git_log

9 git_show

usage : bash submission.sh hash:file

For HEAD :Displays content of file in terminal if file doesnt exists, throws error.
For hash :Displays content of file in that particular commit(Part of hash is sufficient)

10 Search

usage : bash submission.sh search

This **search** command allows users (likely TAs) to search for a student's marks by their unique roll number in the **main.csv** file.

1. Prompts for Roll Number:
   - Reads the roll number to search from the terminal and stores it.
   - Converts the roll number to uppercase for case-insensitive searching.
2. Search for Student:
   - Calls the search_rollno function (defined in **submission.sh**) to search for the roll number in **main.csv**.
   - It uses awk to search the CSV file case-insensitively based on the provided roll number.
   - If found, the entire line containing the student's data (Roll Number, Name, marks) is stored in the **search_result** variable.
3. Display Results:
   (a) If a search result exists (**search_result** is not empty):
      - Prints the header line from **main.csv**
      - Prints the line containing the student's data retrieved from the search.

```
(base) pushpakumar@pushpa-kumar py % bash sub.sh search
Enter roll_no to search marks: 23b0983
Roll_Number,Name,midsem,quiz1,test,Total
23B0983,pushpa,a,23,33,56
```

Figure 8: search_result is not empty

   (b) If no search result is found:
      - Prints an error message indicating the entered roll number is invalid.

```
(base) pushpakumar@pushpa-kumar py % bash sub.sh search
Enter roll_no to search marks: 24b0983
24B0983 is invalid roll number
```

Figure 9: search_result is empty

11 Analyse

usage : bash submission.sh analyse

This **analyse** command analyses marks of students and performs some tasks

1. Data Collection:
   - Iterates through each file except **main.csv** and finally store marks in **marks_to_analyse**

```
midsem 07 16 14 24 10 45
quiz1 23 7 4 23
test 33 11 21
```

Figure 10: example of marks_to_analyse

   - Data Analysis:
     - Creates a TSV file **analyse.tsv** to store analysis results.
     - Writes a header row to **analyse.tsv** containing: "Exam", "Mean", "Median", "Standard_deviation", "Max", "Min".

Exam	Mean	Median	Standard_deviation	Max	Min
midsem	19.33	15.0	12.64	45	7
quiz1	14.25	15.0	8.81	23	4
test	21.67	21.0	8.99	33	11

Figure 11: example of **analysis.tsv**

     - Calls the Python script **analyse.py** and **grades.py** to perform calculations ,grades and generate graphs.
   - Python Script:(**analyse.py**)
     - Reads data from marks_to_analyse file and Iterates through each line (representing an exam)
     - Calculates
       * Average (mean)
       * Median
       * Standard deviation
       * Maximum mark
       * Minimum mark

       and appends it to **analyse.tsv**
     - Two types of graphs
       * Generates a bar graph representing the distribution of marks for the all exams.
       * Generates a bar graph representing the student's marks in all exams.
   - Python Script:(**grades.py**)
     - It gives grades and some comments based on their performance and stores it in **grades.tsv**

grades

Roll_Number	Name	midsem	quiz1	test	Total	Grade	Rank	Comment
23B0983	pushpa	45	23	33	101	AA	6	Excellent performance! Keep up the good work.
22B1003	Saksham Rathi	16	23	a	39	CC	5	Your performance is not satisfactory. You need to put in more effort and attend all classes.
23B0108	Ramesh	24	a	a	24	CC	4	Your performance is not satisfactory. You need to put in more effort and attend all classes.

Figure 12: example of **grades.tsv**

- User Interaction:
  (a) Presents the user with two options for further analysis:
    i. Analyze based on exam name
    ii. Analyze based on student roll number
  (b) Depending on the user's choice:
    i. If exam name is chosen:
      – Lists all available exam names (excluding **main.csv**).
      – Prompts the user to enter an exam name.
      – Finds the corresponding analysis image (PNG) and opens it based on the user's operating system (macOS, Linux, Windows).

```
(base) pushpakumar@pushpa-kumar py % bash sub.sh analyse
1) exam name
2) student roll number
3) anything else to quit
Analyse based on :1
Exams available: midsem, quiz1, test,
enter name of the exam :midsem
```

Figure 13: example of **analysis** based on exam name

    ii. If student roll number is chosen:
      – Prompts the user to enter a roll number.
      – Searches for the roll number in **main.csv** (case-insensitive).
      – If the roll number is found, opens the corresponding analysis image (PNG) based on the user's operating system.

```
(base) pushpakumar@pushpa-kumar py % bash sub.sh analyse
1) exam name
2) student roll number
3) anything else to quit
Analyse based on :2
enter roll number of a student :23b0983
```

Figure 14: example of **analysis** based on roll number

    iii. If an invalid option is chosen, the script exits.