CP367: Assignment 6 – Winter 2024

Due on April 8, 2024 (Before 11:59 PM)

This is an **individual** assignment, and we will try to practice the concepts of threads using C language.

General Instructions:

- For this assignment, you must use C language syntax. Your code must compile using make **without errors.** You will be provided with a Makefile and instructions on using it.
- **Test your program thoroughly with the GCC compiler in a Linux environment.**
- If your code does not compile, **then you will score zero**. Therefore, ensure you have removed all syntax errors from your code.
- **GitHub Classroom-based** repository will be used to keep daily track of your work and its split with your partner. Make sure to follow the steps described at the end of the document to create your repository.
- **Gradescope** platform will be used to upload the assignment file(s) for grading. The link to the Gradescope assignment is available on Myls course page. For submission, connect GitHub Classroom repository and select the branch master to submit on Gradescope. **Make sure your file name is as suggested in the assignment; using a different name may score zero. The question will include parts that would be auto graded and manually graded.**
- Please note that the submitted code will be checked for plagiarism. Submitting these .sh files would confirm that you have not received unauthorized assistance in preparing the assignment. You also confirm that you are aware of course policies for submitted work.
- Marks will be deducted from any questions where these requirements are not met.
- Multiple attempts will be allowed, but your last submission will be graded before the deadline. Instructors reserve the right to take off points for not following directions.

GitHub Repository Creation Instructions:

To keep track of the assignment work, we will be using a GitHub Classroom-based repository. To submit a group assignment project on GitHub Classroom, follow these steps:

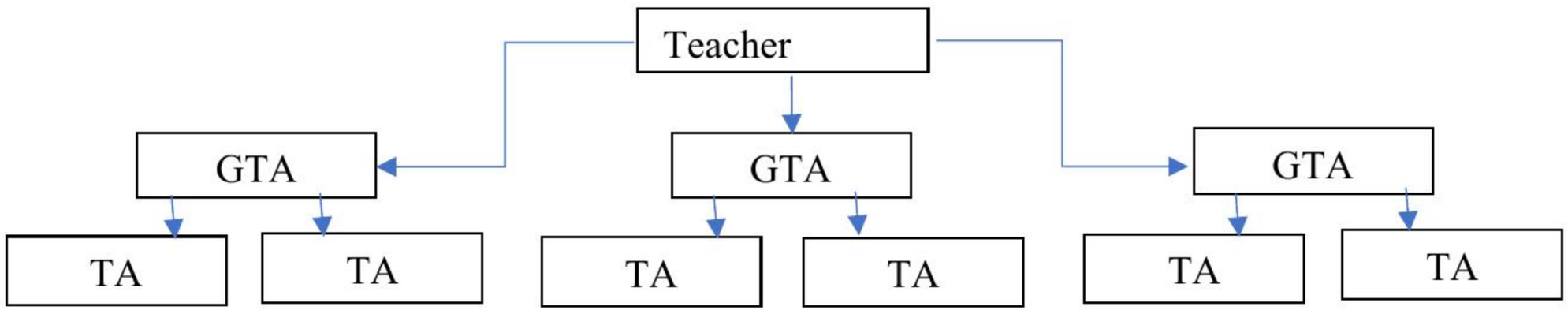
- Make sure that each member of the group has a GitHub account.
- Use the link https://classroom.github.com/a/SSGeGny3 to access the assignment.
- Once you have accessed the assignment, you will see a list of the student names. **Carefully select your name** and click next. You will see a green button that says, "Accept this assignment." Click on the button to create a repository for yourself.
- After creating the repository (by default, Private and keep it as is), you must clone it to their local machine using Git. To do this, navigate to the repository's page and click the green "Code" button. Copy the HTTPS link and use it to clone the repository. You can use GitHub Desktop on your local OS to manage the GitHub repository better.
- Work on completing the assignment and regularly pushing your work onto GitHub. Your activity and contributions will be graded based on the repo push count. If your GitHub account is missing, **you will get zero for the assignment.**
- Keep pushing your changes daily to GitHub so I can check your progress. Once the assignment is complete, push the changes to the repository using Git.
- When you are ready to submit the assignment, navigate to the assignment page on Gradescope and click the "Submit assignment" button. This will prompt you to select the repository that you want to submit. Select the repository that your group created.
- Now, confirm that you want to submit the assignment. You will be asked to provide a comment explaining your submission. Provide relevant details and click on the "Submit assignment" button.

- In the final step, visit Gradescope and select the project assignment. It will ask you to connect to your GitHub account. Once authorized, you can select the repository you were working with and select the branch master to submit on Gradescope.
- That's it! Your assignment has now been submitted to GitHub Classroom. I will be able to review your submission and provide feedback.

Question 1

Create a C program (name it "course_average.c") that finds the average grade of the course assignments. The teacher gives two assignments for every chapter. You must calculate the average grade for each assignment in all the chapters.

- The program should have a teacher process that reads the grades ("sample_in_grades.txt" grades file is available under the Assignment 1 section on Myls) of all assignments of all the chapters and creates a two-dimensional matrix of grades.
- Teacher process then creates a "GradTA" process. Each "GradTA" process takes care of solving a chapter.
- Each "GradTA" process would create "TA" **threads** so that each thread TA can work on one assignment. The "TA" **thread** would calculate and print the average for that assignment. The process tree is shown below:



The input file "sample_in_grades.txt" contains data for three chapters and two assignments for each chapter (six columns). The first two columns are grades for two assignments for Chapter 1, the following two columns are the grades for two assignments for Chapter 2, and the last two columns are the grades for two assignments for Chapter 3. The grades file contains grades for 10 students.

Use the makefile to compile the program written above. The instructions for using the makefile and its contents are provided on the Myls course page. The other implementation details are at your discretion, and you are free to explore them.

To invoke the program, first use the command: `./course_average sample_in_grades.txt` in the terminal OR use the command: `make runq1` via `makefile`.

The expected output for Question 2:

```
Assignment 1 - Average = 10.200000
Assignment 2 - Average = 11.600000
Assignment 3 - Average = 13.700000
Assignment 4 - Average = 13.600000
Assignment 5 - Average = 9.400000
Assignment 6 - Average = 11.600000
```

Note: When submitting the source code files for Question 1, name them like:

- `course_average.c`