CS/SE2340 Term Project - due Wed, Nov 20, 11:59pm

FALL SEMESTER 2024 - Sections 006, 502

INSTRUCTOR: DR. ALICE WANG

Instructions:

Assembly code should be in a .ASM file. Use and Include SysCall.asm in your ZIP file. Provide a PDF report explaining the code breakdown and how to run your code.

Submit all of your work in a ZIP file to eLearning by the due date. If you are not familiar with the ZIP file format you can find out more about it from this link.

Note: name your ZIP files for homework submission as follows:

FinalProject_<FirstName>_<LastName>.zip, e.g. FinalProject_Jane_Doe.zip

IMPORTANT: do not use another archive format, e.g. RAR, because the grader will not be able to see your files, and you will get 0 points.

Topic

The topic for the term project this semester is the ***Math-match*** game, option “Multiply Easy”**.**

For a feel of the game visit:

https://www.mathsisfun.com/games/math-match-game.html

then choose the option “Multiply Easy”.

Note: MARS does not have a graphic capability, so an ASCII characters-based board is sufficient.

Minimum requirements for the program:

- The game is for one player. The program provides the game environment for the user.
- The game board is displayed using ASCII characters (e.g. , -, +, and |) is the minimum requirement. Creative ways to display the board, e.g. with graphics, will earn extra credits.
- Display includes
    - A counter that displays the number of unmatched cards remaining
    - A timer that shows the time elapsed
    - A win screen that reads “Well Done! You finished in xx:xx” where xx:xx is the player’s time
    - A way to start a new game or exit gracefully

Extra credits will be given for:

- Graphic (10 pts) and sound (5pts)

These extra credit features MUST be **documented in your report and explained well** to unlock these extra credits.

Submission Requirements:

At the end of the semester by the due date, **EACH student** must submit a ZIP file containing

1. A **project report** (in PDF format) covering

a) a description of the program,

b) the challenges that you and your partner had and how did the team overcome them,

c) what you have learned by doing the project,

d) a discussion of algorithms and techniques used in the program, e.g. how to display the board, how to check and remember a move from the user? How does the program work?

e) contributions of your partner (peer evaluation),

f) any suggestions you may have (optional).

Make sure that you write **what you have learned by doing this project** and **peer evaluation** parts individually.

2. A **short video clip** demonstrating the program in action. (If the video is too big you can post it on a website, e.g. YouTube, and submit the link). The video should have audio narration explaining the behavior of your program.

3. All **MIPS assembly language modules** that are needed to run your program.

IMPORTANT: **The program must be developed as multiple modules (i.e. multiple .asm files), each module implements one part of the game. If the program is implemented as one file points will be severely deducted.**

4. A **user manual (PDF)** on how to run and how to use the program.

Obviously, items 2. 3 and 4. are the same for you and your partner, but item 1 must be prepared by each of you individually (some sections of the report, e.g. a) and d), can be shared).

How will points be given?

1. First, the implementation (70pts) and documentation (30pts) of the program will be graded based on the requirements listed above.
2. Then the peer evaluation section of the project report (section e) will be taken into consideration when assigning points to each student. If your partner reported that you did not contribute to the project, you may get ZERO pts for this assignment

You should find a partner and get together as soon as possible to discuss the requirements and decide on how to break down the task to assign to each student. Feel free to contact the instructor or TA if more information is needed.

Happy coding!