Ho Chi Minh City University of Technology
Faculty of Computer Science and Engineering

Computer Architecture

Assignment - Second Semester - 2020-2021

Notes:

- Students are requested to submit the MIPS program(s)/source code (.asm files) to the elearning no later than **Sunday, May-16, 2021**. All files need to be compressed into one .zip (**.rar is not accepted**) file before submitting. Assignments must be done individually.
- Students have to demonstrate program(s) on MARS MIPS on **Monday, May-17, 2021 at room A5-106.2**. Students not show up during the demonstration time will get 0 for assignments.
- **Similarity less than 20% in MIPS code is allowed. In other words, you will get 0 if your answers are similar to an another student more than 20%.** We will use the Stanford MOSS system to check the similarity (https://theory.stanford.edu/~aiken/moss/).
- The report should not contains code. Students should present the algorithms as well as the idea in your implementation.

Students follow the following instructions to finish your work.

1. Write a C/C++ program that:
   - Allow user to input the dimension of two 2D matrix <A> and <B> (the matrices may be square or not).
   - Read the input value from file "A.txt" and assign for the matrix <A>.
   - Read the input value from file "B.txt" and assign for the matrix <B>.
   - Read the input value from file "result.txt" and assign for the the matrix <result>, the height of the matrix <result> is the height of the matrix <A>, the width of the matrix <result> is the width of the matrix <B>.
   - Calculate matrix <golden_result>, where <golden_result> = <A> x <B>.
   - Compare the matrix <result> and <golden_result> then print the rate of difference.

   Note: The number of value that are stored in "A.txt", "B.txt", "result.txt" are always equal to the size of their corresponding matrix.
2. Write a MIPS **procedure** to perform read/write file. The input arguments satisfy the following order:
   (a) Name of the file where the data are read from or written to.
   (b) Argument that identify the behaviours of the procedure (read/write).
   (c) The array where the data are read from (read from array and write to file) or are write to (read from file and write to array).

   Function: Read/Write file.
   Return value: None
3. Write a MIPS **procedure** to perform dynamic allocation for an array. The input arguments satisfy the following order:
   (a) The number of elements of the array.
   (b) The size of each element of the array (in bytes).

Function: Dynamically allocate a space based on input requirements.
Return value 1: If total requested size exceed 65536 bytes, return an error code (-1). Otherwise, return success code (0).
Return value 2: The first address of the allocated array.

4. Write a MIPS **procedure** to perform matrix multiplication. The input arguments satisfy the following order:

   (a) The matrix A.

   (b) The matrix B.

   (c) The matrix result.

   Function: Calculate the product of matrices A and B and store in matrix result.
   Return value: If the dimension of matrices A and B does not satisfy the matrix multiplication condition, return error code (-1), else return success code (0).

5. Write a MIPS **program** of matrix multiplication that:

   - Allow user to input the dimension of multiplier and multiplicand matrices.
   - Dynamic allocate space for multiplier and multiplicand matrices.
   - Dynamic allocate space for product matrix.
   - Randomly generate integer value for multiplier and multiplicand matrix.
   - Print the value of multiplier matrix into "A.txt" file, and multiplicand matrix into "B.txt" file.
   - Calculate the product of multiplier and multiplicand matrices and store it in product matrix.
   - Print the value of product matrix into "result.txt" file.
   - During the program, if there are any error, print the error code then exit the program.

Evaluation

- Students complete the requirements.
- Students explain clearly their code.
- Students can implement properly the procedures.
- Students can use appropriately the registers.
- The results of the multiplication are exact.

Question	Score
1	1 pt
2	2 pts
3	2 pts
4	3 pts
5	2 pts
Total	10 pts

Table 1: Score for each question

——————the end——————