# EECS 111
# System Software
# Spring 2016
# Project 1: Fork & Writing shell scripts

Due Date (April 21st, 2016 11:59 PM)

## Instructions

This assignment has an incomplete matrix multiplication code and a shell script. In this assignment, your job is to understand how to use shared memory for inter process communication and complete the code.

The first step is to read and understand the partial code we have written for you. This matrix multiplication creates multiple child processes to divide workloads. It also provides a partial shell script to execute the matrix multiplication code multiple times with different arguments.

You can unzip the project file by using this command: `tar -xvf ./eecs111-proj01.tar.gz`. Then you will see following files:

1. common.c
2. common.h
3. Makefile
4. matmul.c
5. run.sh

The files that you need to complete are `matmul.c` and `run.sh`. `matmul.c` is the main code for matrix multiplication. It creates a $N \times N$ matrix and multiple child processes to perform matrix multiplication. `run.sh` is the shell script that executes compiled binary file multiple times with different arguments. You can run the shell script by using following command: `bash ./run.sh`.

You can build the project code by using following command: `make`. Also, it is suggested to remove all the compiled object and binary files after you modify the source code. You can remove the object and binary files by using following command: `make clean`. However, when you modify the shell script file, you don't need to use `make` and `make clean`. After modifying the shell script, you can run the script.

Complete source code will provide following output:

```
~~$ ./matrixmul 4 2
Child process (0)...
Child process (1)...

Input matrix1:
83      86      77      15
93      35      86      92
49      21      62      27
90      59      63      26

Input matrix2:
40      26      72      36
11      68      67      29
82      30      62      23
```

```
67      35      29      2

[PASS] Data correct!!
Output matrix:
11585   10841   16947   7283
17321   10598   17041   6525
9084    5507    9562    3853
11157   9152    15093   6452
```

The number could be different because the numbers are randomly generated. Completed source code and shell script will produce following output when you run the script:

```
Execute matrixmul 4 1
Execute matrixmul 4 2
Execute matrixmul 4 4
Execute matrixmul 8 1
Execute matrixmul 8 2
Execute matrixmul 8 4
Execute matrixmul 16 1
Execute matrixmul 16 2
Execute matrixmul 16 4
Total 21 children are used to perform matrix multiplication..
```

## Grading

Your project code will be automatically graded. There are two reasons for this:

1. A grader program can test your code a lot more thoroughly than a TA can, yielding more fair results.
2. An autograder can test your code a lot faster than a TA can.

Of course, there is a downside. Everything that will be tested needs to have a standard interface that the grader can use, leaving slightly less room for you to be creative. Your code must strictly follow these interfaces.
Since your submissions will be processed by a program, there are some very important things you must do, as well as things you must not do. For all of the projects in this class...

1. Do not modify `Makefile`, in any case. You don't need to add any file to complete this project.
2. Only modify `matmul.c` and `run.sh` according to the project specifications. We will also being using our own shell script file.

## Submission

You will need to use EEE dropbox for final submission. The deadline for uploading the files is the project deadline:

- Make a directory named `{Student ID}_A1_matrix` in your home directory and copy all the files into that folder.
- Zip the directory that contains all the files. Make sure the directory name and the zip file name are `{Student ID}_A1_matrix`.
- Command to compress: zip -r `{Student ID}_A1_matrix.zip` folder_to_compress
- Log into UCI EEE dropbox with your user name and password.
- Upload the zipped file to **EECS111 Assignment 1** directory in the UCI EEE dropbox.

## Task

I. (70%) Complete `matmul.c` file. After you complete this file, the completed code can calculate matrix multiplication by creating child processes.

II. (30%) Complete `run.sh` file. After you complete this file, the shell script will run matrix multiplication multiple times with different configuration.

## Note

If you have any question regarding the project, please send an e-mail to following address: haeseunl@uci.edu. You MUST cc the professor when you send an e-mail.