**CECS 326-01**

# Operating Systems

## Matthew Zaldana (ID 027008928)

## Assignment 4

### Due Date: 10/26/2021

### Submission Date: 10/25/2021

# Program Description

1. The program is supposed to create 3 child processes that write to a shared memory segment. Each child process calculates seats left for “some event” in its respective class structure pointer. However, they all write to the shared memory segment. Due to the nature of the shared memory segment, how there exist three child processes all writing to the same memory, and the “randomness” of decrementing the number of seats from the random seed generator, each time the program is run, the output is different. That is the main gist of the program – that each run is different and its due to these features in the program. Shmp1.cpp is the main program and waits for all child processes to run, which are executed in Shmc1.cpp. Afterwards, the program ends.

2. The registration.h file is the header file for a structure called CLASS. It holds all the variables necessary for the structure that the other two programs use. Shmp1.cpp is considered the master file. It creates process ids, a process numbers, shared memory segment ids and forks 3 children processes. It then waits until each child process terminates in the wait\_and\_wrap\_up function to remove the shared memory read-write permissions for all child processes to then terminate the program. Shmc1.cpp executes the seats idea. It stores the process name, id, and all the other arguments passed from when a child process was created in Shmp1.cpp. It has a function called sell\_seats that instantiates a random seed generator that sells the number of seats the child process contains and outputs the current value at that moment to the screen. All the while, the child process is also writing to the shared memory when printing out these seats. Once it finishes, the program detaches the shared memory segment from the process and terminates its execution.