**CECS 326-01**

# Operating Systems

## Matthew Zaldana (ID 027008928)

## Assignment 6

### Due Date: 12/7/2021

### Submission Date: 12/9/2021

# 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 in a bus in its respective bus class structure pointer. However, they all write to the shared memory segment. The point of this assignment, however, was to use unnamed semaphores so that each process didn’t have to use a different semaphore; instead, they all used one semaphore to which they wrote and share processes amongst themselves. The semaphore was initialized in the parent given the semaphore existing in the bus structure and destroyed in the parent as well when wrapping up the program. The child utilized the sem\_wait function given the bus pointer as a parameter as a reference to what semaphore to wait on.

2. The parent program initializes the semaphore which is created in the BUS structure and used as an address location reference throughout the parent. For this, it is important to input the correct parameter values for sem\_init so that the function could correctly use the thread processing sharing methods for the semaphore, which is why both values were set to 1. Just as the previous assignment, the seats would sell calling the child program, which would sell seats in the sem\_wait call to the semaphore which locks the semaphore value from the bus pointer, allowing the value to keep decrementing when selling seats. The child program uses this function to call the bus pointer which is in the share memory so that it has access to the semaphore. Sem\_post unlocks the value, also in the child, incementing its value from 0 to 1, the shmc3.cpp finishes. When this occurs, shmp3.cpp destroys the semaphore in the wait\_and\_wrap\_up function, and the program ends.