OS Project 2 – Semaphores Report

Bryce Henderson

brycehenderson257@ou.edu

**ABSTRACT**

In this assignment, I took my code from the previous assignment where four different processes increased a value sum until it reached 1100000 and modified it using semaphores.

**CS Concepts**

Concepts utilizes in this assignment:

* fork(): creating multiple processes
* shared memory between the processes
* semaphores that allows processes to access shared memory safely

**Screenshots**

The following are screenshots of my code running

1.

A screen shot of a computer program

Description automatically generated

2.

A screen shot of a computer program

Description automatically generated

3.

A screen shot of a computer

Description automatically generated

**Analysis**

Unlike the previous assignment, the code runs correctly with the correct result of 11000000 being the final sum at the end. It should also be noted that it takes a little bit of time for the program to run on the gpel machine, around 2-3 seconds. After many attempts, the order in which the processes finished was always the same. I believe that this is due to random chance and if enough runs were executed, it is possible to see change in this order.

**Conclusion**

Using the provided code, this assignment caused very little issues. At one point, I encountered errors when initializing the semaphores, but I just had to change parameter from NSEMS to 0. After solving that issue, it was just a matter of using the wait() function before altering the memory and the signal() function when a process was finished with it.