**PROJECT 2: Semaphore**

To give students an opportunity to experiment with process synchronization mechanisms.

Author Name: Subhrajyoti Pradhan

Instructor Name: Dr. Korzhova

UID: U79333962

Email: spradhan1@mail.usf.edu

**OBJECTIVE:**

The purpose of this project is to give students an opportunity to experiment with process synchronization mechanisms.

**PROCEDURE:**

1. Create 4 processes using fork
2. Processes share variable called total
3. Each process increment variable by 1 100000,200000,300000 and 500000 times respectively.
4. After all the children have finished, the parent process should release the shared memory and terminate.

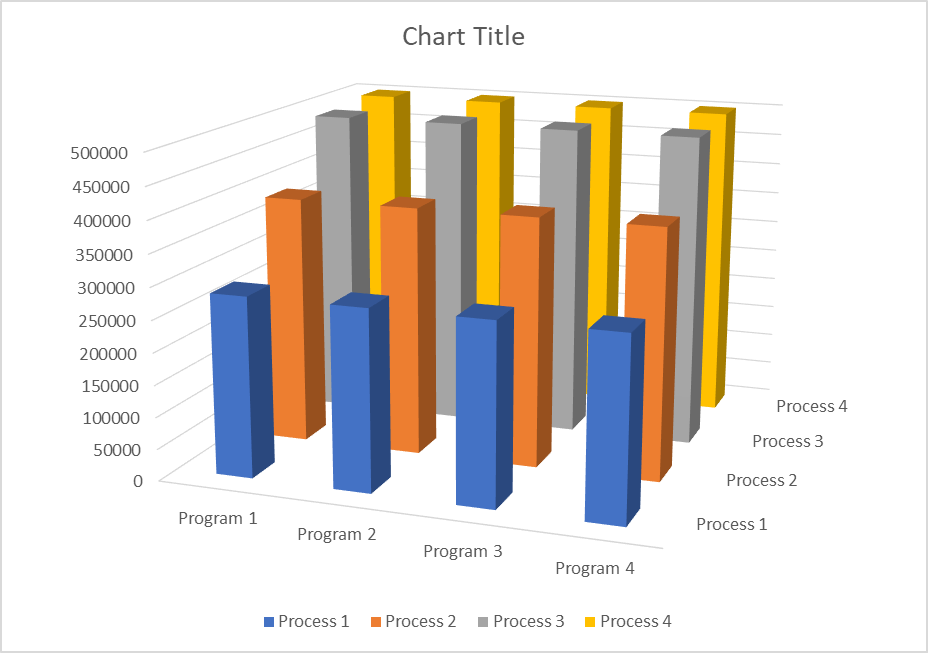
**LANGUAGE USED:**

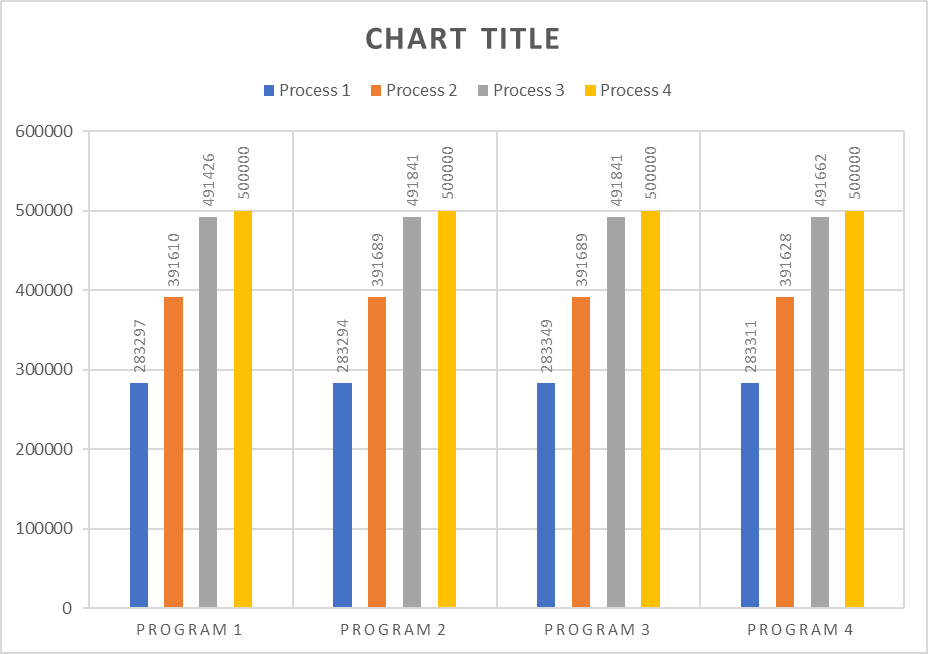
C and compiled in g++ using MobaX Term

**SAMPLE OUTPUT**

**OBSERVATIONS:**

1. The program counter for child executes past the parents assigned range
2. Ex – Process 1 counter ends at 283297
3. The total number of processes are shown in the last process number (11000000).





**ANALYSIS:**

This is an example of parallelism. The processes execute the program at the same time and hence increase the speed of the program greatly. To put the data in shared memory, the I got access to shared memory after checking a semaphore value, wrote the data, and then reset the semaphore to signal to the server that data is waiting. This caused a seamless transition between different processes while working on the same variable and exiting the program upon completion without any great issues.