# PROJECT 1: Shared Memory

To introduce students to the concept of shared memory

Author Name: Subhrajyoti Pradhan

Instructor Name: Dr. Korzhova

UID: U79333962

Email: spradhan1@mail.usf.edu

**OBJECTIVE:**

The purpose of this project is to introduce students to the concept of shared memory and the problems that can occur if shared memory is not protected adequately.

**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:**

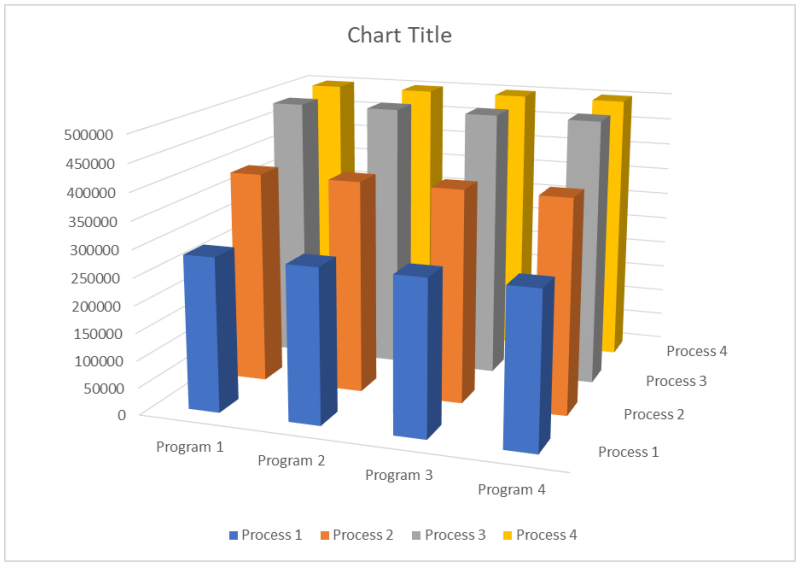
C and compiled in g++ using MobaX Term

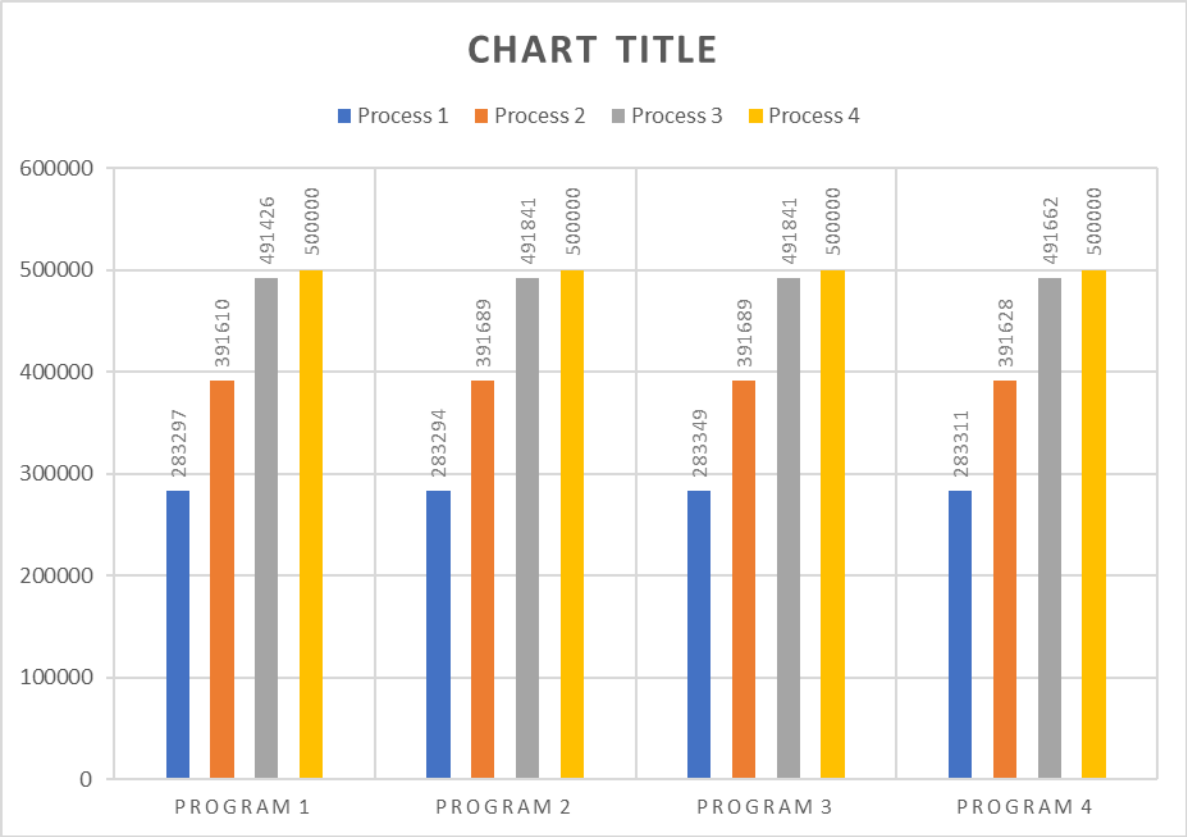
## SAMPLE OUTPUT

```
[spradhan1@itn1 ~]$ ls
a  a.out  candy.cpp  helloworld.cpp  output.txt  process.c
[spradhan1@itn1 ~]$ gcc process.c
[spradhan1@itn1 ~]$ ./a.out
From Process 1: counter = 283297
From Process 2: counter =  391610
From Process 3: counter = 491426
From Process 4: counter = 500000
Child with ID 26512 has just exited.
Child with ID 26513 has just exited.
Child with ID 26514 has just exited.
Child with ID 26515 has just exited.
                End of Program
[spradhan1@itn1 ~]$ gcc process.c
[spradhan1@itn1 ~]$ ./a.out
From Process 1: counter = 283294
From Process 2: counter =  391689
From Process 3: counter = 491841
From Process 4: counter = 500000
Child with ID 27143 has just exited.
Child with ID 27144 has just exited.
Child with ID 27145 has just exited.
Child with ID 27146 has just exited.
                End of Simulation
[spradhan1@itn1 ~]$ ./a.out
From Process 1: counter = 283349
From Process 2: counter =  391667
From Process 3: counter = 491662
From Process 4: counter = 500000
Child with ID 27713 has just exited.
Child with ID 27714 has just exited.
Child with ID 27715 has just exited.
Child with ID 27716 has just exited.
                End of Simulation
[spradhan1@itn1 ~]$ gcc process.c
[spradhan1@itn1 ~]$ ./a.out
From Process 1: counter = 283311
From Process 2: counter = 391628
From Process 3: counter = 491522
From Process 4: counter = 500000
Child with ID 27917 has just exited.
Child with ID 27918 has just exited.
Child with ID 27919 has just exited.
Child with ID 27920 has just exited.
```

## OBSERVATIONS:

1. The program counter for child executes past the parents assigned range
2. Ex – Process 1 counter ends at 283297






**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.