Semaphores Project 2

Karshan Arjun
COP 4600: Operating Systems
University of South Florida, USA

Abstract

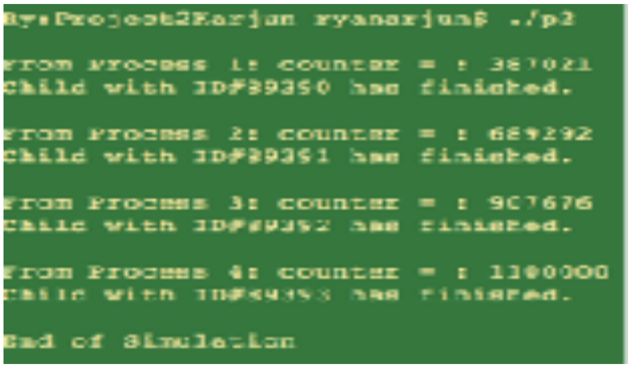
The purpose of this project is to understand the synchronization of semaphores and how they operate in an operating system. The usage of the feature's wait() and signal() are given to processes that will increment a shared segment up to a value of 110,000.

Design

The design of the program is to use two buffers that will wait and signal a process function's critical section when it is ran. In the processes critical section is where the increment of the shared value it being updated. Semaphore functions semwait() and semsignal() are used to initiated the semaphore design.

In the main function the semget() will create a semaphore and then set the value id from the function to system call semctl(), which is used to perform the operations of the semaphore. Once the processes are ran, the semaphore is then deallocated.

Results



```
Ry:Project2Karjun ryanarjun$ ./p2

rrom Process 1: counter = : 367021
Child with ID#39350 has finished.

rrom Process 2: counter = : 689292
Child with ID#39351 has finished.

From Process 3: counter = : 907676
Child with ID#39352 has finished.

From Process 4: counter = : 1100000
Child with ID#39353 has finished.

End of Simulation
```

Performace

The performance of using semaphore is having 0m1.5s on the user mode and 0m6.00s on the kernel mode.