Operating Systems Assignment2Report

10/16/2016

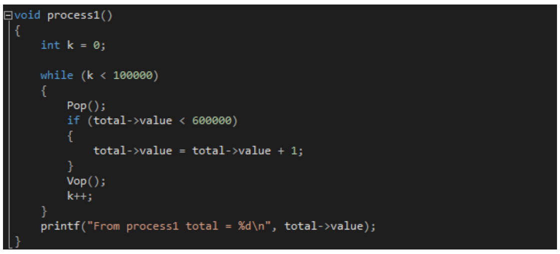
Jinhao Chen

Project Objective:

The purpose of this project is to give us an opportunity to experiment with process synchronization mechanisms. In this assignment, each of processes will share a variable called "total". Each will increment the variable “total” by one to 100,000, 200,000 and 300,000 respectively. In this project, we are tried to modify project1 to protect the critical section using semaphores. After all the children have finished, the parent process should release the shared memory and terminate.

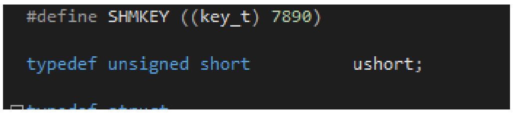
Overview:

In our project1 ass1.c, we already had three processes, and each process independently tries to increase the content of the shared memory location from 1 to a certain value by increments of one. Therefore we just need to create the semaphores to protect the critical section.



```
void process1()
{
    int k = 0;

    while (k < 100000)
    {
        Pop();
        if (total->value < 600000)
        {
            total->value = total->value + 1;
        }
        Vop();
        k++;
    }
    printf("From process1 total = %d\n", total->value);
}
```

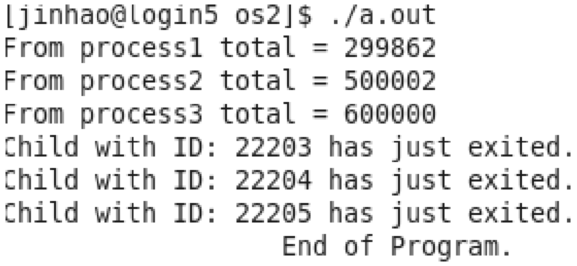
In the previous picture, the code I added was that one if statement and two call function for each process. The reason to add if statement is because our total values should be less than 600000. In addition, the pop and vop functions are tried to call the semaphores to protect the critical section. In project2 instruction, I added all the code inside of this project. Also I added one line code when I compile it using internet compile, it shows cannot recognize ushort, therefore I added one more code to avoid this error:



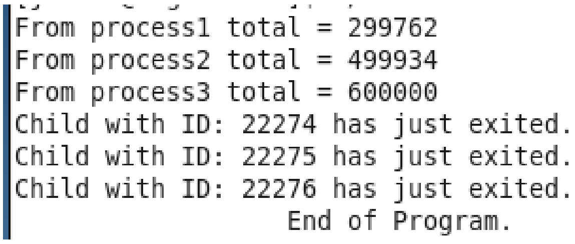
```
#define SHMKEY ((key_t) 7890)

typedef unsigned short          ushort;

typedef struct
```

Conclusion:

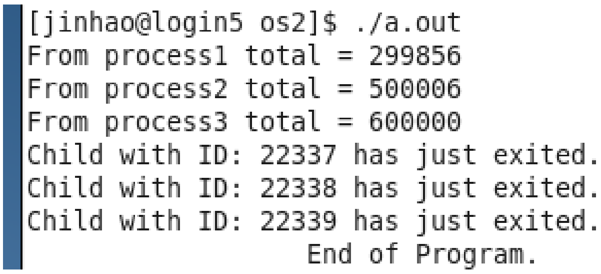
I tested the program several times, and the following images are the results on Linux. In my opinion, it seems that the result of total number for each process looks correctly, because now we had semaphores to protect the critical section, in project 1 we run all process at the same time, but this project we had semaphores to make it asynchornized. Therefore the result should be correct.



```
[jinhao@login5 os2]$ ./a.out
From process1 total = 299862
From process2 total = 500002
From process3 total = 600000
Child with ID: 22203 has just exited.
Child with ID: 22204 has just exited.
Child with ID: 22205 has just exited.
                         End of Program.
```



```
From process1 total = 299762
From process2 total = 499934
From process3 total = 600000
Child with ID: 22274 has just exited.
Child with ID: 22275 has just exited.
Child with ID: 22276 has just exited.
                         End of Program.
```



```
[jinhao@login5 os2]$ ./a.out
From process1 total = 299856
From process2 total = 500006
From process3 total = 600000
Child with ID: 22337 has just exited.
Child with ID: 22338 has just exited.
Child with ID: 22339 has just exited.
                         End of Program.
```