Lab One Report

Angel Camacho 10/4/2020

Username: acamacho

CS 474

My out put from three runs:

```
--------------------------------

Executed: process1
Result: 100000
Finished Child PID: 7649
Executed: process2
Result: 200000
Finished Child PID: 7650
Executed: process3
Result: 300000
Finished Child PID: 7651
Executed: process4
Result: 500000
Finished Child PID: 7652

--------------------------------

Executed: process1
Result: 100016
Finished Child PID: 7654
Executed: process2
Result: 200000
Finished Child PID: 7655
Executed: process3
Result: 300000
Finished Child PID: 7656
Executed: process4
Result: 500000
Finished Child PID: 7657

--------------------------------

Executed: process1
Result: 100000
Finished Child PID: 7659
Executed: process2
Result: 200007
Finished Child PID: 7660
Executed: process3
Result: 300000
Finished Child PID: 7661
Executed: process4
Result: 500000
Finished Child PID: 7662

--------------------------------
```

It is not specified how long or short the report is supposed to be.

In this project we were introduced to the concept of shared memmory, and the problems that can occur. We created four processes which share a common variable named "total". An important key to note here is the fact that withing each process we looped to increase our common variable that I previously mentioned of. After each function is called I then printed the total value.

Important key details would include making sure that the newly created chilld only. This is done by the following:

```
    if((pid1 = fork())==0){
        process1();
        exit(0);
    }
```

As we can se heare we make sure theat when the fork function is executed the child pid and parent pid should be difrent thus not leting any more children execute it again.

Another key detail would be the fact that I would get diffrent values each time I runeed my code this is due to the fact that memory is beaing waited on for the next proccess in order to resume. Other than that I put a lot of comments which explain several sections of my code.