

Project 2.1 Report

Keller Sedillo-Garrido

10/17/2022

Problem Description:

For this programming assignment, we were tasked to create 4 child processes that share a value called total. Each of the 4 processes will increment the value from one to 100,000, 200,000, 300,000 and 500,000 respectively. At the end of each process, the program will print out the counter and end the share processes.

Analysis:

After running the program multiple times, we see an interesting trend of data.

```
gcc -o shareMemory shareMemory.c
./shareMemory

From Process 1: counter = 68282.
From Process 2: counter = 224666.
From Process 3: counter = 236824.
From Process 4: counter = 560081.
Child with ID: 395235 has just exited.
Child with ID: 395236 has just exited.
Child with ID: 395237 has just exited.
Child with ID: 395238 has just exited.
```

```
gcc -o shareMemory shareMemory.c
./shareMemory

From Process 1: counter = 49112.
From Process 2: counter = 207463.
From Process 3: counter = 374148.
From Process 4: counter = 444284.
Child with ID: 395224 has just exited.
Child with ID: 395225 has just exited.
Child with ID: 395226 has just exited.
Child with ID: 395227 has just exited.
```

Project 2.1 Report

Keller Sedillo-Garrido

10/17/2022

```
gcc -o shareMemory shareMemory.c
./shareMemory

From Process 1: counter = 138818.
From Process 2: counter = 274839.
From Process 3: counter = 379991.
From Process 4: counter = 508332.
Child with ID: 395213 has just exited.
Child with ID: 395214 has just exited.
Child with ID: 395215 has just exited.
Child with ID: 395216 has just exited.
```

```
gcc -o shareMemory shareMemory.c
./shareMemory

From Process 1: counter = 66747.
From Process 2: counter = 238708.
From Process 4: counter = 378359.
From Process 3: counter = 410328.
Child with ID: 395202 has just exited.
Child with ID: 395203 has just exited.
Child with ID: 395205 has just exited.
Child with ID: 395204 has just exited.
```

```
gcc -o shareMemory shareMemory.c
./shareMemory

From Process 1: counter = 83246.
From Process 2: counter = 261644.
From Process 3: counter = 277128.
From Process 4: counter = 529696.
Child with ID: 395181 has just exited.
Child with ID: 395180 has just exited.
Child with ID: 395179 has just exited.
Child with ID: 395182 has just exited.
```

Looking at these results, I have a few thoughts on the shared processes. One, we see that each of the processes don't count properly. We see that some of the processes end before they reach their target. For example, some of the Process 1 have less than 100,000. Second we see that the processes were also executed at different times. For example, you would normally see 1,2,3,4. But one of the processes ended

Project 2.1 Report

Keller Sedillo-Garrido

10/17/2022

with 1,3,2,4. This means that processes that are not properly protected will overlap one another.

Conclusion:

Without proper protection of our values, we see that our processes will mess up and create inconsistent answers as an outcome. If we don't take good care with how we share our memory, we could botch our output.