CS 474 – Project 1

Processes and Shared Memory

Antonio Maldonado
Computer Science
New Mexico State University
Las Cruces, New Mexico
m21tony@nmsu.edu

Description

In this program, four processes will be created, all sharing a variable named 'total'. Each process will loop and increment by one until the variable reaches 100,000, 200,000, 300,000, and 500,000 in each process, respectively. After they have all finished and printed, the parent process will release the shared memory and terminate the program.

Results

After running the program and evaluating the results, we can see that we never get the same output from the processes. Five tests were ran, and although the processes didn't increment the 'total' variable counter to be the exact number that we wanted, it was very close:

	RUN 1	RUN 2	RUN 3	RUN 4	RUN 5
PROCESS 1	74442	82298	71248	164577	177116
PROCESS 2	210216	202273	204832	181763	219981
PROCESS 3	288122	329403	300685	350578	342235
PROCESS 4	476270	525650	497630	519019	496322

Analysis

I believe that the results seen are because of the CPU having to save some data in cache memory. The shared memory is very fast, but the access of it in such high speeds can cause the CPU to cache

data instead of storing it in main memory. Since some of the data can be false, or become incoherent, some of the data is not shared properly. With this kind of program, some data degradation is expected, as reflected in the results with each run not being consistent or as expected.

Output: Screenshots

```
newton cs474/project1> ./project1

From Process 1: Counter = 74442
From Process 2: Counter = 210216
From Process 3: Counter = 288122
From Process 4: Counter = 476270

Child with ID 7946 has just exited.
Child with ID 7947 has just exited.
Child with ID 7948 has just exited.
Child with ID 7949 has just exited.
End of program.
```

Run 1

```
[newton cs474/project1> ./project1

From Process 1: Counter = 82298
From Process 2: Counter = 202273
From Process 3: Counter = 329403
From Process 4: Counter = 525650

Child with ID 12298 has just exited.
Child with ID 12299 has just exited.
Child with ID 12300 has just exited.
Child with ID 12301 has just exited.
End of program.
```

Run 2

```
[newton cs474/project1> ./project1

From Process 1: Counter = 71248
From Process 2: Counter = 204832
From Process 3: Counter = 300685
From Process 4: Counter = 497630

Child with ID 12351 has just exited.
Child with ID 12352 has just exited.
Child with ID 12353 has just exited.
Child with ID 12354 has just exited.
End of program.
```

Run 3

```
[newton cs474/project1> ./project1

From Process 1: Counter = 164577
From Process 2: Counter = 181763
From Process 3: Counter = 350578
From Process 4: Counter = 519019

Child with ID 12357 has just exited.
Child with ID 12358 has just exited.
Child with ID 12359 has just exited.
Child with ID 12360 has just exited.
End of program.
```

Run 4

```
[newton cs474/project1> ./project1

From Process 1: Counter = 177116
From Process 2: Counter = 219981
From Process 3: Counter = 342235
From Process 4: Counter = 496322

Child with ID 12362 has just exited.
Child with ID 12363 has just exited.
Child with ID 12364 has just exited.
Child with ID 12365 has just exited.
End of program.
```

Run 5