COP 4600 Operating System

Project #1 (Shared Memory)

September 9, 2018

Project Objectives:

The purpose of this project is to introduce students to the concept of shared memory and the problems that can occur if shared memory is not protected adequately.

Total points Available: 100

Due: September 9 at 11:59 pm

Project Description:

In this assignment, you will create **4 processes**. Each of these processes will share a variable called "total". Each will increment the variable “total” by one 100,000, 200,000, 300,000, and 500,000 times respectively. Make sure that only the newly created child calls the function “process#()”

After **all the children have finished**, the parent process should release the shared memory and terminate. Use the **"wait"** function so that the parent knows precisely when each of the children finishes. The parent should print the process id of each child as the child finishes execution. Then it should release shared memory and print "End of Program".

You need to run program several times and analyze your observations (write report).

Sample output

From Process 1: counter = 270547.
From Process 2: counter = 347860.
From Process 3: counter = 400001.
From Process 4: counter = 500000.

Child with ID: 2412 has just exited.
Child with ID: 2411 has just exited.
Child with ID: 2413 has just exited.
Child with ID: 2415 has just exited.

End of Simulation.

Submitting your assignment

- Submission via Canvas Assignment.
  - It is your responsibility to submit these assignments in a timely fashion.
- All files should be zipped together.
- There should be a readme file explaining in detail the exact steps to be taken to compile and execute the code files and the title page
- Testing of this work should be done only on the CS lab machines. Please make sure these machines are not locked up due to your code. The execution for grading purposes will be done on the lab machines.
- In case of any code errors, partial credit may be offered based on the code and documentation.
- A report that presents the performance evaluation of your solution.
  - The report should be properly formatted (an academic format style, such as ACM or IEEE being preferred) and contain quantitative data along with you analysis of these data.

Late Submission Policy

- Late work will be not accepted.

Grading Criteria:

- Minus 90% if code does not compile. Minus 70% if it compiles but does not run.
- If the code compiles and runs, further deductions will be made for the following:
  - Minus 40% if 4 children are not created.
  - Minus 40% if the processes are not cooperating processes.
  - Minus 30% if the children fail to modify the shared variable.
  - Minus 20% if parent ends without waiting for all children to exit.
  - Minus 10% if parent does not release shared memory before ending.
  - Minus 10% if the report is not written
  - Minus 10% if children do not print out their results.
  - Minus 10% if parent does not print each time a child finishes.
  - Minus 5% if your program is not commented and not formatted appropriately
  - Minus 3% if your name is not included in comments on the top of your source code

Development Environment

You may write your program using any available editor Nano, Emacs, Vi or whatever editor you are most comfortable with, BUT, it must compile with gcc and be executable on one of the CS machines:

To login to these machines remotely, download PUTTY (for Windows, Linux users skip this step) by going to: http://the.earth.li/~sgtatham/putty/latest/x86/putty-0.58-installer.exe
Then after the download, execute PUTTY. Also you need to download and install Junos pulse from USF VPN (https://www.net.usf.edu/vpn/Windows/ = for windows or https://www.net.usf.edu/vpn/index.php = for other OS types). Click PUTTY and enter one of the lab machines for the Host Name. Their hostnames are osnode[01-16].csee.usf.edu. Then enter login name and password (your netid)
Hints:

Build your project in an incremental fashion. Attempt to meet each objective before moving on to the next.

The UNIX commands, needed for the project, are given at the end of lecture 4.

Find more information about these command and options used by them by using UNIX manual or by simply using man command.