

Programming Assignment 2
Creating Processes and Threads and Communication between Processes
Due: On Canvas 2020

Goal: Get experience creating and running concurrent processes and threads.

Part 1:

Description:

You will create processes, run processes in parallel, and pass information between processes.

The data to be processed are text strings, lines of CSV separated values, we wish to sort these.

1. Get any one of these data sets:
`http://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php`
(all quakes last 30 days)
or
`http://www-odi.nhtsa.dot.gov/downloads/flatfiles.cfm`
(and get the FLAT_RCL file (zip) which are all vehicle recalls in the US (the RCL.txt is the "meta"-data (schema) description of the data)
- 1a. If you sort the earthquake data, sort by latitude, in ascending order, if you sort the recalls sort by year of recall and manufacturer (first year, then manufacture name).
2. Either: sort the data by hand (not a good idea),
or write a two loop (bubble or insertion) sort program to sort the data. (Do not use a "better" (faster) sorting method, the purpose is to use a "slow" sort method, but if you really want to, use much, much bigger data.)
3. "Instrument" your program (time it).
4. Create a program that will, in turn, run multiple processes "concurrently" (at the same time, employing all of your CPUs and CPU Cores, using "fork()" and "exec()" (there are several variants of exec (execl, execv, etc.)
Please do not use "threads" (yet).
5. Do the sort, again, in parallel for 2 concurrent processes, then 4, and finally 10 processes.
6. Instrument those sorts (like above).
7. How will you pass data (parts of the array) to each process (IPC)? (Files, pipes, shared memory, message queues?)
8. You may (not required) to use a menu to select number of processes, size of data, etc.

Hints:

Creating Processes (fork, exec):

https://ece.uwaterloo.ca/~dwharder/icsrts/Tutorials/fork_exec/

Shared memory:

<https://stackoverflow.com/questions/5656530/how-to-use-shared-memory-with-linux-in-c>

<http://users.cs.cf.ac.uk/Dave.Marshall/C/node27.html>

<https://opensource.com/article/19/4/interprocess-communication-linux-channels>

Part 2:

Description:

You will create a process and threads, run in parallel, and pass information between threads.

You will use synchronization operations to avoid overwriting shared resources.

As in Assignment above:

1. Use the same data set, as assignment 2.
2. Sort, using the same methods from part 1, (write a two loop (bubble or insertion) sort program to sort the numbers in ascending order.
3. "Instrument" your program (time it).
4. Create a program that will, in turn, run multiple threads "concurrently" using a kernel level threading system.
(there are several options: Pthreads, Java, C/C++)
5. Do the sort, again, in parallel for 2 concurrent threads, then 4, and 10 threads.
6. Instrument those sorts (above).
7. How will you pass data (parts of the array) to each thread?
(How will you synchronize sharing resources such as memory?)
8. You may (not required) to use a menu to select number of threads, size of data, etc.

Hint:

Lawrence Livermore Lab thread tutorial:

<https://computing.llnl.gov/tutorials/pthreads/>

includes thread creation, management, and using mutexes

Please, Submit ONLY to Canvas.

All work must be your own, you may reference web sites, books, or my code but You MUST site the references.

You must submit this lab, working (or partially) by the due date.

You may be asked to demonstrate this lab, working (or partially) to the GTA after the due date.

Your program should be well commented and documented, make sure the first few lines of your program contain your name, this course number, and the lab number.

Your comments should reflect your design and issues in your implementation.

You should address error conditions.