
ID1206 HT24

Review Lab for Module #1

Deadline: 2024-11-15

1. Process pairs: Create multiple processes and communicate via FIFOs

Implement a program based on the following specifications:

- Use `fork()` to create 1 child process.
- Let pairs of processes communicate via one or more FIFOs.
- Use `wait()` on the master to gracefully terminate the set of processes.

2. Test global sum performance

Use the code shown in the Review Question #8 for Module #1 (“Multithreaded summation performance”). Let $\Delta t(N, N_{\text{thread}})$ be the program’s execution time for array length N and number of threads N_{thread} . Perform measurements with 1 and 2 threads and estimate the array length \tilde{N} where

$$\Delta t(\tilde{N}, 1) \equiv \Delta t(\tilde{N}, 2).$$

Include in your report a plot showing $\Delta t(N, N_{\text{thread}})$ as a function of N . Document relevant properties of the processor that was used for this experiment. Hint: Check the content of the pseudo-file `/proc/cpuinfo`.

Hints:

- Time measurements can be performed using any of the following functions
 - `gettimeofday`
 - `clock_gettime(CLOCK_THREAD_CPUTIME_ID, ...)`.
- Since a single summation completes within a short period of time, consider to measure the time needed for $M \gg 1$ repetitions.

3. Master-workers: Create multiple processes and communicate via messages

Implement a program based on the following specifications:

- Use `fork()` to create ≥ 1 child processes.

- Let the parent and all its child processes communicate via 1-2 message queues.
- Define a protocol that is used by the parent to send commands to the workers.
- Bonus: Implement a queue used by the children to send messages to the parent.
- Use `wait()` on the master to gracefully terminate the set of processes.