

Homework 4
Computer Systems Organization II, V22.0202.003 Spring 2005, Professor Yap

Out: Mar 8. Due: Mar 22 (after Spring Break).

INSTRUCTIONS:

- This homework in Interprocess Communication is worth 30 points.
- Please read questions carefully. All handed in work must your own (even if you have discussed with other students)
- Please start your programming part immediately! Feel free to send me and the grader emails when you run into problems.

• This programming homework continues to explore the use of concurrency to solve problems. We now want to solve the many gcd problem of the last homework using threads instead of processes. We will be using POSIX threads, so you might want to reread lecture 7. We will give some additional hints on pthreads.

- OVERVIEW. Your main program is called `gcd.th.c`. The input to `gcd.th.c` is a set of pairs of numbers. If there are $k \geq 1$ pairs, then the **main thread** will spawn k children threads, called **gcd threads**. But the main thread also spawns one **worker thread** who will do the actual numerical computation (namely, modulo operation)

Each gcd threads will compute the GCD of one pair of numbers. Thus, the gcd thread must submit pairs (a, b) of integers to the worker thread who will compute and return the modulus $a \bmod b$. When a gcd thread has computed the GCD of its pair, it prints the result and exits. The main thread will wait for all the gcd threads to exit. When all the gcd thread has exited, the main thread will send a "thread cancellation" command to the worker thread to kill it. Then the main thread exits.

- COMMUNICATION. The main communication mechanism is to through shared variables, but protected by mutex variables and conditional variables.
- Finally, you are to create several set of runs (using various sets of input pairs), and give the timing for these runs.

Furthermore, we want you to do the same runs for the solution in the previous homework. Please compare the timings and draw some conclusions about the relative speeds of process creation/communication versus thread creation/communication.

- WHAT TO HAND IN: a single tar file containing a Makefile file, README file, and all necessary programs. You must give your timings and explain your experiments in the README file.

I should be able to duplicate your experiments by typing "make time".