

EC527: High Performance Programming with Multicore and GPUs

Programming Assignment 4

Objective

Learn about and practice multithreaded programming.

Prerequisites (to be covered in class or through examples in on-line documentation)

HW – We assume a basic multiprocessor with shared address space.

Programming – Basics of PThreads: create, passing parameters, join, sync, barrier.

Assignment

Lab setup

We are back to writing high-performance code, this time using eng-grid. For this week you will probably want to use the lab machines for most of your tests, because they have eight cores. The tutorial can be done on any multicore processor with a standard Linux environment.

To do: the tutorial in **pthread_tutorial.pdf**

To be handed in: as specified in the “Tasks” -- answers to questions, modified code, etc.

Task 17 utilizing `test_crit.c` involves manipulating the timing in which the thread executes without touching the logic of the thread. In order to force threads to overwrite and cause an error in the balance calculation, on the lab machines you might need to use the

```
int nanosleep(const struct timespec *req, struct timespec *rem);
```

method as the `sleep(int time);` method used during other parts in the lab might cause too much of delay, (and thus no overwriting).

This method uses the `<time.h>` header and `timespec` struct that we should be familiar with from previous labs. For more detail on how to use it the following resource should be helpful:

<http://linux.die.net/man/2/nanosleep>