CSCI 251: Concepts of Parallel and Distributed Systems

Alvin Lin

September 18th, 2017

Topics

- Complete Bitonic Sort
- Shared Memory
- Processes and Threads
- POSIX (P. Threads)
- Project 1 Problem

Shared Memory

The idea with shared memory is that you allow multiple processes to share the same memory. Computers may have separate memory process access, but use a logically shared address space. Concurrently, all processes can read from the memory space with no problems, but processes that write to the memory space must write atomically and serially. A mutual exclusion (mutex) lock is necessary for this to happen. A semaphore is a generalized mutex that can be used to control access to memory by processes.

Processes and Threads

Processes:

- A sequence of instructions

- Process ID, User ID
- Program instructions and PC
- Stack
- Heap
- Registers
- File descriptors
- Libraries
- Mechanisms for message passing

Threads:

- Created by a process
- Shares process resources (file descriptors, libraries)
- Stack pointer
- Registers
- Lightweight
- Can be executed independently
- Scheduled for execution by the OS

Example code:

```
for (row = 0; row < n; row++)
  for (col = 0; col < n; col++)
    c[row][col] = dot_product(get_row(a, row), get_col(b, col))
    # it is feasible to create a thread here to do this
```

Project 1

Refer to MyCourses for details on Project 1

Reminders

Professor Mohan Kumar:
mjkvcs@rit.edu
https://cs.rit.edu/~mjk

Rahul Dashora (TA):
rd5476@mail.rit.edu

You can find all my notes at http://omgimanerd.tech/notes. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech