Final Exam Review

Science

Computer Science

CS 450: Operating Systems
Michael Lee <lee@iit.edu>

ILLINOIS TECH | College of Computing

Topics

- Concurrency
- Threads
- Locks
- Semaphores
- I/O
- HDDs

Midterm Exam Logistics

- Monday 5/1 8AM-10AM in SB 104
- Up to 3 pages of typed or handwritten notes permitted on exam
- Calculators ok, no phones/other devices

Concurrency

- What is it?
- Why do we want it?
  - Improved CPU & I/O utilization
  - Increase in performance (speed-up)
- How is it implemented?
  - Time multiplexing vs. Parallelism
  - Processes vs. Threads

Threads and Threading models

- How are threads different from processes?
- What does each thread require (as implementation)?
- Threading models (implementation, pros/cons)
  - User-level (green) threads
  - Kernel-level threads
  - Hybrid threading model

Limits of Parallelism

- Amdahl’s & Gustafson’s laws
  - What are they?
  - When are they applicable?
  - What are their ramifications?

Race conditions & Critical Sections

- What is a race condition?
  - What conditions are necessary for them to exist?
- What is a critical section?
  - How might we deal with them?

Locks & Locking strategies

- What is a lock & how is it used?
- Locking strategies (description, pros/cons)
  - Coarse grained
  - Fine grained

Implementing locks

- Spinlocks & Ticket locks
  - Why do we need h/w support? What do we need?
  - Where are they implemented?
- Pros/Cons of spinlocks
- What can we build on top of them?

Sleep / Wakeup

- Why do we have these mechanisms?
- How are they implemented/used in xv6?
  - Why do we still need spinlocks?

(No Pthreads!)

Semaphores & Synchronization

- What are the basic semaphore rules?
- Essential patterns:
  - Rendezvous, Mutex, Multiplex, Generalized Rendezvous
- Basic problems:
  - P/C, R/W (Lightbulb pattern), Dining Philosophers
- Applying these patterns!

Practice problems!

- In the Little Book of Semaphores, study:
  - The “Dining Savages” problem
    - A “scoreboard” pattern for synchronizing threads
  - The “River Crossing” problem
    - A pattern for “matching” different thread types

Concurrent programming paradigms

- What is the “default” model, and why is it hard to work with?
- Software transactional memory
- Actor model (Message passing)
- Even-driven programming (Asynchronous)

I/O devices

- Basic I/O device model & protocol
- Protocol variants
  - Basic protocol
  - Polled vs. Interrupt driven
  - Programmed I/O vs. Direct Memory Access
  - Special instructions vs. Memory-mapped I/O

HDDs

- Basic HDD geometry and considerations
- Seek/Rotate/Transfer access mechanism
  - Disk throughput computation
- Disk head scheduling algorithms