CS202 - Midterm Review

Admin

1. In-person, during class on Thursday
   Please don't be late, scheduled for 75 mins

2. Closed book. Can use a cheat sheet
   - One letter paper sheet, can write/printf on both sides
   - 10 pt font/legible handwriting

Topics we covered

Exam rationale: Spend some time revisiting & thinking about the topics covered in the class, labs, readings, hw.

1. Introduction
   - What the class covers
   - History — safe to ignore
Processes & Calling Conventions

→ What is a Process
   → Process Control Block

→ How Processes are Created
   → Process

→ Registers
   → What they are

→ Caller-Saved vs Callee Saved

→ The Stack
   → Grows Down

→ PushQ a, g
  \%esp = \%esp - 8
  *\(\%esp\) = a, g

→ PopQ

→ Calling a Function

   → Passing Arguments — Conventional

   → The Call Instruction

→ The Function Prolog
   → Sets Up a Stack Frame
   \%ebp = Base Point
- Local Variables: Where They Live.

→ Returning From A Function

- The Function Epilog
  - Destroys Stack Frame.
  - Restores Callers Stack Frame.
  - Callee Saved Registers?

  → Return Value
  - Not

  → User & Kernel Mode: Traps

(3) Concurrency

- Threads
  - Differences From Processes

- Concurrency & Why It Is Hard

- Mechanisms To Control Concurrency
  - Mutexes
  - Condition Variables
  - Semaphores
  - Spin Locks
- Monitors & How to Use Them
- Deadlock
- Fairness, Starvation & Other Concerns

4) Scheduling

- Process State Machine

Init → Runnable ← Running → Terminated

- Pre-emptive vs Non-Pre-emptive Scheduling
  - Yield

- Metrics
  - Turnaround Time
  - Output Time
  - System Throughput
  - Resource Utilization
  - Fairness

- Policies
- FCFS/FIFO
- STCF
- Round Robin
- Priorities
  - Way Useful
  - Strict Priority
  - MLFO

- Lottery/CFS
  - Just high-level idea

6 Therac-25
  - Expectation: Read the paper
  - Understand it at a high-level.

6 Virtual Memory - I
  - Goal
  - Virtual vs Physical Addresses
int f() {
    int x = 5;
    return x;
}

int main() {
    int *x = f();
    g(x);
}

void g(int x) {
    int y = x;
    printf("%.d \n", y);
}