

Getting Started Project 2

Part 1 – Tracing System Calls

- Write an empty C program
 - Strace it to see how many syscalls it produces
- Write a small C program
 - Strace it to see how many syscalls it produces
- Iteratively change the small program until it is 10 calls more than the empty program

Part 2 – xtime Module

- Setup simple procs hello world module
- Get the value in xtime
 - Display in proc file
- Store value on each proc read
- Take difference of two time values when last read exists
 - Similar to project 1's etime command
 - Display in proc file

Part 3 – Elevator

- Write simple procs module
- Design how the pieces should fit together
 - Don't just start writing code immediately
- Write framework
 - Can use OO techniques covered last week
- Add in lists
- Add procs output
 - Do early for debugging
- Add system calls
 - Do early for outside communication
 - Use in testing driver
- Add in threading / locking
 - Make sure code is robust before starting
- Make scheduler more complex
 - If you want extra credit

Pacing

- Week 6
 - Install kernel
 - Do part 1 and part 2
 - Do some initial design on part 3
- Week 7
 - Write elevator framework
 - Add in linked lists
 - Setup proc output
- Week 8
 - Setup system calls
 - Add threading and locking
- Week 9
 - Finish threading
 - Work on scheduler
 - Last minute tweaks
- Week 10
 - Due!