CS230/330: Project 2 Supplement

Handed out: Tuesday, January 30, 2001
Due: Friday, February 9, 2001, 11:59 p.m.

1 Assignment

Due to the extended deadline of project 2, this handout describes additional features that must be added to your project 2 handin. None of these features are particularly difficult to implement (well, provided that you have memory management and context switching working).

Before starting this part of the assignment, please review the Project 2 handout. Your kernel should be loading two processes 'init1' and 'init2' and context switching between them on clock interrupts before continuing.

1. Implement a queue data structure and use it to implement the kernel's ready queue. Modify your kernel so that 'init1' and 'init2' are placed on the ready queue during kernel startup.
2. Modify your TRAP_CLOCK handler so that it schedules the next process on the ready queue. The currently running process should be placed on the end of the ready queue.
3. Implement the `Yield()` system call. This simply removes the running process and puts it at the end of the ready queue.
4. Implement the `GetPid()` system call.
5. Implement the `Delay()` system call.
6. Modify your kernel so that it only runs the kernel idle process when there is no other useful work to do (i.e., the ready queue is empty and there is no other runnable process). Your kernel should never run the idle process if there is a user program that can run instead.
7. Implement the `DisplayWrite()` system call so that user processes can write to the display.
8. Implement the `PenRead()` system call so that user processes can read from the pen input. To implement this, your TRAP_PEN interrupt handler should queue up pen events into a kernel buffer. The `PenRead()` system call should then read data from this buffer. If no pen events are available, `PenRead()` should block the calling process until an event arrives.