CS39002: Operating Systems Lab
Spring 2012

Assignment 6
Due: March 15, 2012, 1 pm

In this assignment, you will explore the **/proc** filesystem in linux. The **/proc** filesystem provides a means to get and set various information about the kernel and about particular processes.

You have to first write a C program *SystemInfo.c* that will read the /proc file system and print out the following (with an appropriate message in each case):

1. The number of CPUs in your machine and their clock speed (note that different cores are counted as different CPUs)?
2. the version of Linux kernel running on your system
3. The time in day:hr:min:sec when the system was last booted
4. The average load on the system in the last 5 minutes
5. The total usable and currently free memory in the system
6. The swap partitions and their sizes
7. The time did the CPU spend (over all processes) in the user mode, kernel mode, and in servicing interrupts
8. The number of context switches made by the system so far

Next write another C program *ProcessInfo.c* that gets the following information specific to a process. The program takes the *pid* of the process as a command line argument.

1. The command line with which the process was started
2. The time spent by the process in running and in waiting
3. The time spent by the process in the user mode, kernel mode, and in servicing interrupts
4. **[Extra Credit]** Can you print out the contents of the address space of the process?

In order to answer the above questions, the files of the /proc filesystem that will be relevant for you are *cpuinfo, uptime, loadavg, cmdline, stat, meminfo,* and *schedstat* (except for the Extra Credit question; for that, you need to find out on your own what you need). Some of these files will be under /proc directly, some will be under the directory for the specific process, and some will be under both. You will need to read and understand what is contained in these files from the net and implement the above program. Note that the exact format of the files vary somewhat between different versions of Linux, so you should try to write your program in as format-independent manner as possible (the names of things are mostly standard).