

# CS 340 Fall 2012 Programming Assignment II and Paper

## The Program Due 11/9

Add scheduling for your disk queues, a round robin scheduler for your ready queue and CPU usage accounting to your OS. time accounting

✓ 1 Along with all the information you got in the first program's SysGen, you should prompt the installer (me) for the length (in milliseconds) of a time slice and how many cylinders each disk has.

✓ 2 This program should do *everything* your first program did as well as the following: <sup>end of</sup> While the OS is running the timer (me) signals an interrupt with a "T". If this happens the process in the CPU should be sent to the "back" of the ready queue. All other interrupts and system calls are, as before, handled uninterruptedly and instantaneously. <sup>into</sup>

✓ 3 If a process in the CPU issues a system call the system should ask the timer (still me) how long the process was in the CPU. A call for a disk read or write should prompt for the cylinder as well as all the other information.

1 The disk queue should no longer be a FIFO queue, but one ordered so that the head seek time is minimized while no requests are starved. See "The Paper" for more details.

5 An S,{r/c/d/p} should show all the appropriate information well as the total time each process in the displayed queue has used the CPU, each processes average burst time and the systems average total CPU time of competed processes. When a process terminates its PID and total CPU time are reported to the accounting module (me) and system's average total CPU time is updated.

macosx files! Email source code and compile instructions *before* class on the due date. If you are sending one or two files, there is no need to package them. If you are sending more than two you should package them. Tar and zip are fine. Please make sure that the files unpack into the SAME directory the source file is in. If you are sending c++ code, and I can compile it with `g++ -o ~/temp/run.me *cpp` then I don't need a makefile. If you do supply a makefile, please make sure the output executable is `~/temp/run.me`. Also be aware that I will count the readability of your output in the grade.

A CPU burst does not end with a time slice, it ends with system call  
When a system call is made, you'd have to query how much time is left.

## The Paper Due November 6

In order to decide on your disk operation ordering algorithm you will need to do some research. Your goal is to find a fast algorithm that won't starve any waiting processes.

Research the known algorithms and write a 500-1000 word paper (2-3 pages, double spaced, 1 inch margins, 12 point times roman) analyzing the choices and justifying your choice to implement one of them. I will compare what you choose in the paper to what you implement in the program, but success in that will effect your program grade. This paper will be graded on readability (grammar, sentence and paragraph structure, understandability) completeness and documentation.

Hand in a printed, stapled document at the beginning of class. No cover or title pages, but your name should be somewhere. All your work should be documented, remember that undergraduates may have opinions but are not supposed to know anything. Fact sources should be referenced. I don't care if you use foot-notes, in-notes or end-notes or whether you stick to Chicago, MLA, APA or some other documentation standard, but all the "right" info has to be in the references and bibliography. The bibliography does not count to the size of the paper.