

Lab 3 – Critical Sections

Due: 9:55 AM, Tuesday, March 1, 2005

This week’s lab consists of questions to look at on your own (not to be turned in) and some questions to be turned in. The programming for this lab is considered a “laboratory program” for honor code purposes. You may discuss the programs with others, but what you turn in must be your own work.

Practice Questions

You do not need to turn in answers to these questions. But they could certainly form the basis for potential exam questions.

- Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favors those processes that have used the least processor time in the recent past. Why will this algorithm favor I/O-bound programs and yet not permanently starve CPU-bound programs?
- SG&G 6.3, 6.4

Lab Questions

Answer these questions in a plain text file named `lab3.txt`.

1. List two enhancements that could be implemented in your round robin simulator from Lab 2 that would allow a more realistic simulation or allow the simulation of more sophisticated scheduling algorithms. For each, describe briefly what your enhancement would allow the simulator to do, and how you would go about implementing it. (2 points)
2. SG&G 6.1 (3 points)

Process Interleavings

Write a C program that will list all possible orderings of the machine instructions generated for the critical sections of the Producer-Consumer example from class. Recall that the statements `counter++` and `counter--` actually generate machine code such as

Producer	Consumer
P_1 <code>R0 = counter;</code>	C_1 <code>R1 = counter;</code>
P_2 <code>R0 = R0 + 1;</code>	C_2 <code>R1 = R1 - 1;</code>
P_3 <code>counter = R0;</code>	C_3 <code>counter = R1;</code>

Your program should list all possible interleavings of the statements $P_1, P_2, P_3, C_1, C_2,$ and C_3 . Also have your program print which interleavings produce a correct result (that `counter` has the same value it started with).

Write your program in a file called `interleaving.c`. (5 points)

Submission and Evaluation

- All necessary files should be submitted using `turnin` as a single “tar” file, `lab3.tar`. Include a `Makefile` to allow easy compilation of the C program.