

Homework 8

Due: 7 May 2009

Problem 8.1

Would a process ever want to acquire both a read lock and a write lock on the same resource? Why or why not? How would it/could it go about doing so safely? Does your answer change if there are multiple threads or processes that would want to be doing this?

Problem 8.2

Consider the following use case for a process-scheduling system. Processes that arrive “at time t ” are considered to arrive infinitesimally before time t , and processes p_1 and p_2 are assumed to be in the queue in that order.

Process	In queue at time	Length	Priority
p_1	0	5	5
p_2	0	2	8
p_3	3	5	3
p_4	6	10	4
p_5	11	1	1

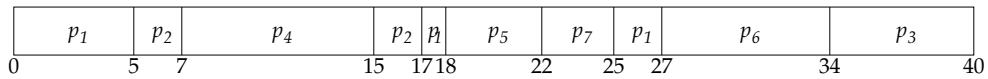
Draw usage charts for the different CPU scheduling algorithms:

- FCFS
- SJF
- SJF-preemptive (aka shortest-remaining-time-next)
- non-preemptive priority scheduling
- preemptive priority scheduling
- round robin

Here, lower numbers mean higher priority: the most VIP process has priority 1.

Problem 8.3

Consider the following processor-usage chart.



It corresponds to the following set of process arrival times.

Process	In queue at time
p_1	0
p_2	5
p_3	6
p_4	7
p_5	18
p_6	19
p_7	20

- What CPU scheduling policy(ies) might give rise to this chart? How do you know?
- Calculate the average turnaround time for the processes according to this CPU allocation.