University of Karlsruhe
System Architecture Group
Gerd Liefländer

Teaching Assistant:
Philipp Kupferschmied

System Architecture 2008/09
Assignment 9

Question 9.1: Design Parameters

1. Why might it be beneficial to run a mixture of CPU-bound and I/O-bound processes instead of scheduling first all CPU-bound processes and all I/O-bound processes afterwards (or vice versa)?
2. With preemptive scheduling, each process/thread is granted a *timeslice* of a certain length, after which it is preempted and another process/thread is chosen by the scheduler. Discuss pros and cons of choosing a short timeslice length vs. choosing a longer timeslice length.

Question 9.2: Concurrency

1. Try to explain the notions *preemptive kernel* and *non-preemptive kernel*.
2. Give examples of *blocking* and *non-blocking kernel operations*.
3. Enumerate some good reasons *not to use semaphores* within *applications*.

Question 9.3: Server and Desktop Scheduling Policies

Five batch processes, A through E, arrive at a computer center at essentially the same time. Their estimated execution times are 15, 9, 3, 6, and 12 minutes, respectively. Furthermore, the processes have externally defined static priorities of 4, 7, 3, 1, and 6, respectively, with 10 being the highest priority. Determine the turnaround time for each process and the average turnaround time for all five processes for each of the following scheduling algorithms. Ignore process switching overhead completely. Explain how you arrived at your answers.

1. round robin (RR) with a time slice length of 1 minute
2. priority-based scheduling
3. first come, first served (FCFS)
4. shortest job first (SJF)

In the last three cases, assume that each process runs until it finishes and that all processes are completely processor-bound.
Before you start to calculate, please try give an estimated order amongst the policies concerning the performance measure *average turnaround time*.

Question 9.4: Scheduling à la SJF

Prove that *shortest job first* (SJF) provides the minimum mean turnaround time among all non-preemptive scheduling algorithms.

Question 9.5: Scheduling in SMP Systems

1. Name some considerations that apply to SMP scheduling but not to single-processor scheduling.
2. Describe the basic ideas behind co-scheduling and gang scheduling in your own words.
3. A multi-processor scheduler can choose to use either a single central ready queue or multiple ready queues (one per processor). Discuss the pros and cons of using a central ready queue.
4. Assume an SMP load balancing algorithm that performs *work stealing* to balance load between the processors: A processor that falls idle can *steal* a process from one of the other processors. In which cases does this work stealing algorithm fail to perform proper load balancing?

Question 9.6: Lottery Scheduling

1. Describe the idea of lottery scheduling.
2. Enumerate possible advantages of lottery scheduling over "classical" priority-based scheduling algorithms.
3. Discuss possible implementations of lottery scheduling. How can tickets be implemented? How can the next process to run be found?