Operating Systems
Semester 2, 2008
Eike Ritter

The University of Birmingham
12 October, 2008

Exercise sheet 2

1. (a) A mono-processor has several processes in the ready-queue:

    - one process plays computer music via the sound card;
    - four processes perform a long compilation;
    - one process is an editing process.

    The process playing the music needs the processor 75% of the time for the music to be of high quality. Assume further that the compilation process and the editors access different disks. Describe the effects of Round-Robin, Shortest-Job-First and priority scheduling (with suitably chosen priorities) on the response time of these processes.

    (b) Assume the music playing process requires a response time of 1ms. Can we use virtual memory for the memory assigned to this process?

2. Suppose that a scheduling algorithm favours those processes that have used the least processor time in the recent past. Why will this algorithm favour I/O-bound programs and yet not permanently starve CPU-bound programs?

3. Show that the Shortest-Job-First algorithm minimizes the average waiting time. (Hint: consider the effect on the average waiting time of an exchange of a process with a short and a long burst time).

4. The solution for the reader-writer problem with different semaphores for reader and writer does not prevent starvation of writers. Modify the solution in such a way that starvation of both readers and writers is not possible.

5. A barbershop consists of a waiting room with n chairs and a barber room with one barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy, but chairs are available, then the customer sits in one of the free chairs and falls asleep. If the barber is asleep, the customer wakes up the barber. If the barber is finished with one customer, he wakes up all customers which are asleep in the chairs. Write a program to co-ordinate the barber and the customers using suitable semaphores. You may assume that there are operations `BarberSleeps` and `CustomerSleeps` which block. Also there are operations `WakeupBarber` which wakes up the barber, and `WakeupCustomers` which wakes up *all* customers which are currently asleep.