Homework 3: Synchronization primitives
Deadline: 05/05/2011. (Due in class)

Several of the problems are from your text book. Please contact me if you do not have the text book.

1. What is the meaning of term “busy waiting”? What other kinds of waiting are there in the operating system? Can busy waiting be avoided altogether?

2. Show that if `semWait` and `semSignal` operations are not executed atomically, then mutual exclusion may be violated. Use the lecture note handout implementation for your answer.

3. The sleeping barber problem: A barbershop consists of a waiting room with n chairs and the barber room containing the 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 occuppied, 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. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers.

4. A file is to be shared among different processes, each of which has a unique number. The file can be accessed simultaneously by several processes, subject to the following constraint: The sum of all unique numbers associated with all the processes currently accessing the file must be less than n. Write a monitor to coordinate the processes.

5. Write a monitor that implements an alarm clock that enables a calling program to delay itself for a specified number of time units (ticks). You may assume the existence of a real hardware clock that invokes a procedure tick in your monitor at regular interval.

6. Consider a system consisting of processes P_1, P_2, ..., P_n, each of which has a unique priority number. Write a monitor that allocates three identical line printers to these processes, using the priority numbers for deciding the order of allocation.