DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF CAPE TOWN.

Operating Systems: Assignment 2 (10.03.2014)

Anne Kayem

March 6, 2014

1 OVERVIEW

In this practical assignment we will be focusing on the problem of synchronisation in an operating system. As discussed in class, classic problems of synchronization play the role of "models" that operating systems designers and programmers use in implementing algorithms to handle access to shared variables.
As a programming exercise we will be working on the classic synchronization problem that is termed *"The Sleeping Barber Problem"*. This synchronization problem is somewhat similar to the bounded buffer model of the producer-consumer problem but different in terms of the handling of the buffer.

2 THE SLEEPING-BARBER PROBLEM

A barbershop consists of a waiting room with n chairs and a barber room with 1 barber chair. If there are no customers to be served the barber goes to sleep. If a customer enters the barbershop and all the 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. If the barber is asleep, the customer wakesup the barber.

3 ASSIGNMENT OBJECTIVES

In this assignment, you will be implementing synchronization algorithms to coordinate the behaviour of the barber and the customers in the Sleeping Barber problem, to avoid situations of deadlock and stravation. You may use either Java, C++, or C for implementation. For marking/grading simplicity please use either a Windows or Ubuntu operating system.

4 Question 1

1. Using the either concept of semaphores or monitors, write a program to coordinate the barber and the customers.

5 Question 2

Modify your synchronization program for the sleeping barber problem to create a situation of deadlock.

6 Question 3

Modify your synchronization program for the sleeping barber problem to create a situation of starvation. (**HINT:** You could use the concept of ready and ready/suspend queues as a basis for your implementation.)

7 Question 4 - Bonus

As an analogy assume that the sleeping barber problem were transposed to an operating system so that customers represent processes, the barber's waiting room represents a ready queue, and the barber's room the CPU. Assuming that the dispatcher is supported by a multilevel feedback queue algorithm, provide implementations to answer the questions that follow:

1. Compute the average wait time for customers when the wait room is full. Use randomization to generate arbitrary integers to represent the estimated times required per customer for hair cuts.
2. Suppose that instead of turning away customers when the wait room is full, we use a prioritization scheme based on time required to provide a hair cut to encourage customers to stay and process the shorter haircut jobs first. Propose an implementation of an algorithm to prevent starvation.