ECES 338 Assignment #4

Due: March 2, 2000

Spring'2000 **(100 points)**

Ozsoyoglu, G.

In this assignment, you will work with monitors. For your algorithm, you can use any pseudo-code (or C or C++) syntax. I will not accept any late assignments after March 8th, at which time I will post the solutions to the assignment.

(35 points) **1)** (*The Roller Coaster Problem*). Suppose there are n passenger processes and one (roller coaster) car process. The passengers repeatedly wait to take rides in the car, which can hold C passengers, C < n. However, the car can go around the tracks only when it is full.

(a) Develop codes for the actions of the car and passenger processes, and develop a monitor-based solution to synchronize them. Explicitly specify any assumptions you make about the model.

(b) Generalize your answer to part (a) to employ m car processes, m > 1. Since there is only one track, cars cannot pass each other; i.e., they must finish going around the track in the order in which they started. Again, a car can only go around the tracks when it is full. Explicitly specify any assumptions you make about the model.

(35 points) **2)** (*The Meeting Problem*) Two kinds of processes, A's and B's, enter a room and "meet" processes of other type. For a monitor-based solution, the definition of "enters' is: the process is allowed into the monitor (i.e., the room). The definition of "meets" we use is: Once a process of one type enters the room, it is considered to have met all the processes of the other type that are in the room. (Another possible definition is: a process of one type enters the room, and starts shaking hands with processes of the other type (in the order of their arrival). Handshaking constitutes an instance of "A meets B". We will not use this definition).

An A process cannot leave until it meets two B processes, and a B process cannot leave until it meets one A process. A process (of type A or B) leaves the room once it has met the required number of the other kind of processes.

(a) Develop a monitor-based solution to implement this synchronization. Explicitly specify any assumptions you make about the model.

(b) Modify your answer to (a) so that the first of the two B processes that meet an A process does not leave the room until after the A process meets a second B process. Explicitly specify any assumptions you make about the model.

(30 points) **3)** (*Drinking Philosophers Problem*) Consider the following generalization of the dining philosophers problem. Given is an undirected graph G. Philosophers are associated with nodes of the graph and can communicate only with neighbors. A bottle is associated with each edge of G. Each philosopher cycles between three states: tranquil, thirsty, and drinking. A tranquil philosopher may become thirsty. Before drinking, the philosopher must acquire the bottle associated with every edge connected to the philosopher's node. After drinking, a philosopher again becomes tranquil. Give a monitor algorithm to this problem that is deadlock-free.