جامعة حلوان
Helwan University
Faculty of Computers and Information

Sheet 4

Department of Computer Science
Data Structures

1 Introduction

One of the main elements in the cellular mobile systems is the Mobile Telephone Service Office (MTSO). It provides interfacing of the mobile system to the public switched telephone network. Many users can use their cellular phones simultaneously, and the MTSO has to handle these competing calls. The FIFO policy is not always fair, since more important calls must be served promptly. For example, if there are some voice calls currently in the system—either waiting for connection or already connected—any emergency call must have a priority over them. In addition, any voice call must have a priority over data calls.[1]

Since this sheet is educational at a very introductory level, we will consider an over-simplified simulation for the prioritization system mentioned above. Assume that the system handles only three types of calls; Data Calls, Voice Calls, and Emergency Calls. Any user starts using his cellular phone and placing a voice call, he should have the least priority in the system with respect to other voice and emergency calls. However, it is fair enough to have higher priority among other data calls. This means that the new voice call should be inserted in the data structure lagging all the existing voice and emergency calls but leading all the existing data calls. Similarly, a new emergency call should be inserted in the data structure lagging all the already-existing emergency calls but leading all other calls. A new data call is always inserted in the structure at the rear.

Generally speaking, in simulating discrete-event systems—as the one in our current application—the new calls arrive and depart the system randomly by utilizing a random number generator. To make things simple, we will require that a new call is placed/dismissed by selecting an option in the menu of the main program.

2 Main Program

Write a simulation program for the system described above. The program should display the following menu.

1. Data Call coming.
2. Voice Call coming.
3. Emergency Call coming.
4. Serve a call.
5. Dismiss calls with least priority.
6. Exit.

Hints: Each call should have the following fields: a tag (D, V, or E), the number of the calling party, and the number of the party to be called. In addition, a Data call has extra one field (number of packets); and a voice call has extra two fields (a boolean variable for showing the caller ID and another boolean variable for roaming option). These fields should be entered for every call.

The option "Serve a call" should display the call at the front of the queue then delete it from the system. The option "Dismiss all of the least prior calls" should delete from the system all of the data calls, if any in the system; otherwise, it should dismiss all of the voice calls. "Exit" should save the contents of the data structure into a file then quits. In addition, when the program starts it should create a new structure and loads it from the same file. This requirement has no practical value; it only helps in saving the simulation information for next run.

You will need three different counters in your main program to keep track of the number of the calls for each call type. This is to enable you inserting the call in the right order as described in Section 1.

[1]Students, who are interested in the subject for future research or projects, may refer to Mohasseb, Y. Z. (1996), *Prioritized channel assignment in cellular radio networks,* M.sc. thesis, Military Technical College, and Yousef, W. A. (1999), *Simulation of Prioritized Channel Assignment Models in Cellular Mobile Radio Networks,* M.sc. thesis, Helwan University.