

## BIL342 Experiment III

---

# Hacettepe University Department of Computer Science and Engineering

## BIL342 Programming Laboratory Experiment III

<b>Subject:</b>	Concurrent and Multithreaded Programming with Linux
<b>Advisors:</b>	Prof. Dr. Ali SAATÇI R.A. Kerem ERZURUMLU R.A. Kıvanç DOĞANAY
<b>Submission Date:</b>	31/03/2006
<b>Deadline:</b>	18/04/2002
<b>Programming Language:</b>	ANSI C

### AIM

The aim of this experiment is to give students the ability to cope with the basic concepts of concurrent multithreaded programming with race conditions on Linux. Within this context you are asked to write a C program with several threads which will manipulate a shared file concurrently.

### PROBLEM DESCRIPTION

In this experiment, you will be simulating operations carried on within the Department of the National Lottery.

Your program should get the lottery data from 2 different data files: Accounts and Transactions data files.

A sample account data file is shown in Figure-1.

```
<Costumer ID>:<Costumer Name>:<Balance>  
1:Kerem ERZURUMLU:1000  
2:Zerrin Ünal:200  
3: ...
```

Figure-1. Sample account data file

Each row of the account data file is related to a single client. Each client has an account with a balance. Balances cannot be less than zero. The balance unit is YTL.

## BIL342 Experiment III

In the transaction data files each row represents a transaction. A sample transaction data file is shown in Figure-2.

```
type1 1 4 5 6 8 12 15 2
type1 LD 1
type1 2 1 2 3 4 5 6 1
type2 1 30
type2 2 50
type3 8 10 12 14 16 18
```

Figure-2. Sample transaction data file

In each row the first field represents the transaction type. For each type of transaction the format of the related line is different. There will be only three types of transaction:

1. **type1:** The line format for this type of transactions can be in 2 different formats:

```
type1 client_ID number1 number2 number3 number4 number5 number6 times
```

```
type1 client_ID LD times
```

where numbers represents the “guess” of the client; times represents the number of successive draws for which the guess is valid. Each “guess” costs 5 YTL and therefore times x 5 YTL will be withdrawn from the client’s account.

LD is a short for “LuckyDip”. When a transaction shows a LD, your program will randomly decide 6 numbers for the client.

2. **type2:** This type of transaction is used to deposit money to an account. The line format is as follows:

```
type2 client_ID amount
```

where amount represents the sum of money to be added to an account with a client\_ID. The amount can be negative if the client has withdrawn money from his account.

3. **type3 :** represents the lottery winning numbers.

```
type3 number1 number2 number3 number4 number5 number6
```

When a type3 transaction occurs; the total sum gathered with type1 transactions will be divided by 4 and each quarter will be shared equally by 6, 5, 4 and 3 right number guessers.

Your program should include 5 threads. As there are 3 types of transactions, there will be 3 different threads executing a particular type of transaction. These threads will get transactions from their own transaction queues and will execute them on the account matrix concurrently. The fourth and fifth thread will be reading transactions files row by row and append each row to the appropriate transaction queue (Figure-3).

You are asked to program your threads in a fully synchronised manner using mutex or semaphores (The use of mutex is the preferred way because they are primarily designed for thread synchronization; semaphores are generally used for process synchronization).

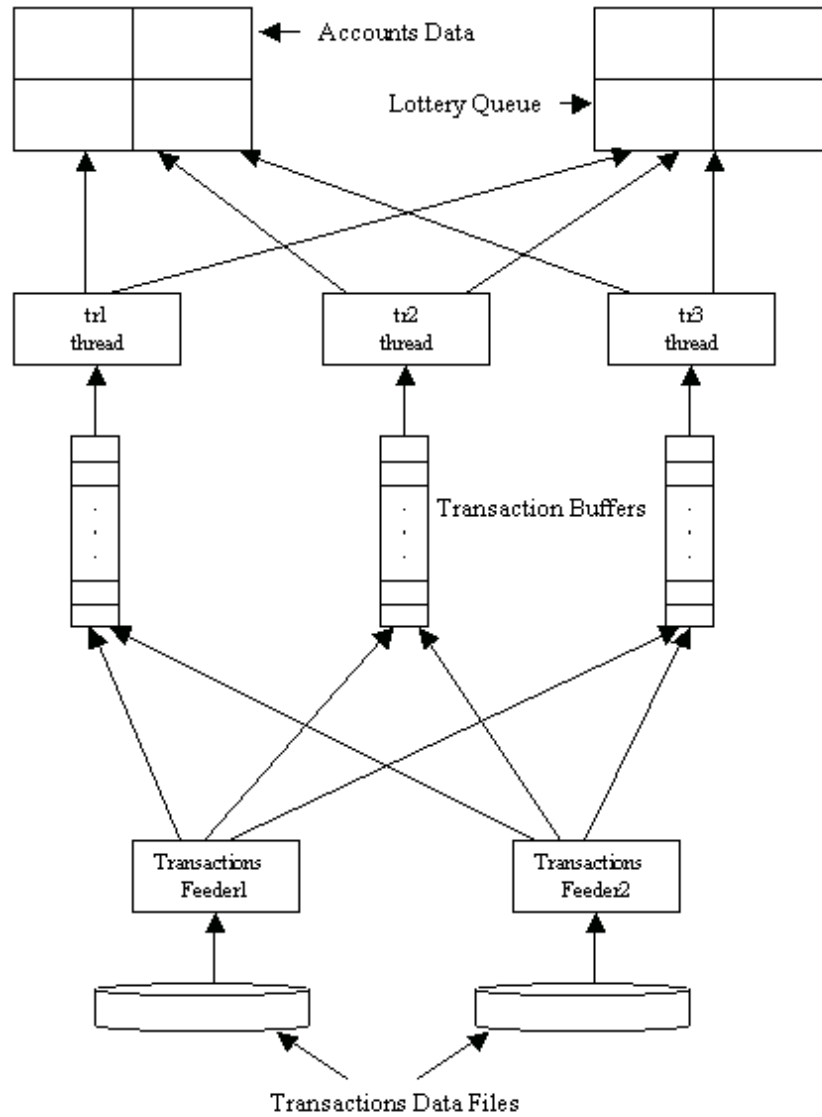
Once you have finished testing the synchronized version of your program, you are also asked to run it without any synchronization and observe the abnormalities by marking synchronization function lines as comment lines.

### USER INTERFACE

Your program will get two arguments from **command line**; filenames of the account and the transaction data files respectively.

Every time a thread completes a transaction it should display a message on the screen. This message should include all the necessary information about the identity of the thread and the completed transaction.

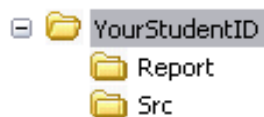
## BIL342 Experiment III



**Figure-3. The execution schema of threads**

### NOTES

1. You are asked to follow announcements made to Courses.Bil342 newsgroup which is located at [nntp://news.cs.hacettepe.edu.tr](mailto:nntp://news.cs.hacettepe.edu.tr) .
2. The soft copy of this paper can be found at  
<ftp://ftp.cs.hacettepe.edu.tr/pub/dersler/Bil3XX/Bil342/05-06/3>
3. Your report and program must be submitted at the same time.
4. You are asked to give both soft and hard copy of your reports. Valid Soft-Copy formats are HTML and PDF.
5. You should submit your work and report on a floppy with the following structure:



6. E-mail submissions are not accepted.
7. Late submissions will not be accepted.

## BIL342 Experiment III

---

8. At the submission your experiments will be checked for common errors. Works with this kind of errors may be refused for grading. Do not leave everything to the last minute.
9. Office hours will be held on Wednesday afternoons.
10. Please send any additional questions to Courses.Bil342 newsgroup. Good Luck