

**CS 3305A: Operating Systems**  
**Department of Computer Science**  
**University of Western Ontario**  
**Assignment 3**  
**Fall 2019**  
**Due Date: Nov 29<sup>th</sup> 2019**

**Purpose**

The goals of this assignment are the following:

- Get hands-on experience in developing mutual exclusion / semaphore / critical section techniques/algorithms.
- Gain more experience with the C programming language from an OS's mutual exclusion / semaphore / critical section perspective.

**Assignment Description**

Using C programming language, you will be developing mutual exclusion algorithm for the following banking scenario. You need to make sure that your mutual exclusion algorithm allows no more than one client accessing the critical section portion of your code at any given point in time. You are allowed to use any mutual exclusion / semaphore related C function calls.

a) Description of the problem is given below:

Let  $A$  represent a set of  $n$  bank accounts, where  $n \geq 1$ . Let  $C$  represent a set of  $x$  clients, where  $x \geq 1$ . Each bank account in  $A$  can either be a *Personal Bank Account* or a *Business Bank Account* shared by all clients in  $C$ . Clients can perform three different types of transactions with each bank account: deposit, withdraw or transfer funds. Depending on the type of bank account, there can be a fee associated with depositing, withdrawing and transferring funds to a bank account. Each bank account also has a limit on the number of transactions that can be performed. Any transaction beyond this limit may still occur with a fee.

Some bank accounts in  $A$  may have overdraft protection. If the bank account has overdraft protection, at the end of a transaction, the account balance can be less than zero. However, an overdraft fee will be charged for every \$500 that the client borrows from the bank to complete the transaction. There is a limit of \$5000 on the overdraft. If the bank account does not have overdraft protection, any transaction resulting in the account balance to be less than zero should not occur.

Initially, there are a set of  $i$  depositors, where  $i \geq 1$ . Clients are not allowed to start any transactions until all depositors deposit the money into the bank accounts.

b) Structure of the input file:

In the following example, there are two bank accounts (a1 and a2) shared by a total of ten clients/users (c1 to c10). The clients are allowed to deposit money into both the accounts, withdraw money from both the accounts, and transfer money between the two accounts. There are also two depositors (dep1 and dep2) who deposit money initially into both the accounts. The clients are not allowed to start their banking activities (i.e., deposit, withdraw, and transfer) until the depositors deposit the money into the accounts. An input file is provided below for illustrative purposes.

```
a1 type business d 0 w 5 t 10 transactions 20 2 overdraft Y 40
a2 type personal d 0 w 0 t 0 transactions 10 1 overdraft N
```

```
dep1 d a1 1000 d a2 3000
dep2 d a1 2000 d a2 2000
```

```
c1 d a1 100 w a2 500 t a1 a2 25
c2 w a1 2500 t a1 a2 150
...
...
c9 w a1 1000 w a2 500
c10 d a1 50 d a2 200
```

Illustration:

```
(i) a1 type business d 0 w 5 t 10 transactions 20 2 overdraft Y 40
```

The above line specifies information related to bank account #1. Account #1 is a business account. There is a fee of \$0 to deposit funds, \$5 to withdraw funds and \$10 when funds are transferred into the account or out to another account. After the 20<sup>th</sup> transaction, there is an additional fee of \$2 for every transaction. This account has overdraft protection and a fee of \$40 will be charged for every \$500 that the account borrows to complete the transaction.

```
(ii) dep1 d a1 1000 d a2 3000
```

The above line specifies the operations performed by depositor #1. Depositor #1 deposits \$1000 into Account #1 and then deposits \$3000 into Account #2

```
(iii) c1 d a1 100 w a2 500 t a1 a2 25
```

The above line specifies the operations performed by client #1. Client #1 deposits \$100 into Account #1, then withdraws \$500 from Account#2, and then Transfers \$25 from Account #1 to Account #2

Your program must use “assignment\_3\_input\_file.txt” for processing all the operations outlined in the input file. Do not assume the input file name will be a command line argument.

c) Structure of the output file:

You must output the balances of each bank account after all the transactions have been performed. Your C program should output results to the screen and into a text file “assignment\_3\_output\_file.txt”. Your output must be in the following format:

```
a1 type business -300
a2 type personal 550
a3 type business 4500
a4 type personal -45
```

**Assignment related technical resources**

Please visit the course website for specific technical instructions and relevant materials. Also, consult Instructor and TAs for any question you may have regarding this assignment. A test case will be provided on the course website for your convenience.