



Computer Architecture and Technology Area

Universidad Carlos III de Madrid

OPERATING SYSTEMS

Lab 3. Multithreading

Bachelor's Degree in Computer Science & Engineering
Bachelor's Degree in Applied Mathematics & Computing
Dual Bachelor in Computer Science & Engineering & Business
Administration

Year 2022-2023

Contents

- 1 Introduction
- 2 Lab Description
- 3 Material
- 4 Delivery

Contents

- 1** Introduction
- 2 Lab Description
- 3 Material
- 4 Delivery

Introduction

- Development of a program that acts as a bank managing account operations from ATMs.
 - The user provides a file with a list of account operations: CREATE, DEPOSIT, WITHDRAW, BALANCE, TRANSFER (**input file**)
 - The system must execute the operations on the indicated bank accounts and display the total balance of the bank (**output**).
- For the calculation of the balance:
 - Load the data from the file into an array in memory.
 - Initiate an N—producers and M—consumers system
 - The producers (ATMs) insert the data from memory into the shared circular buffer.
 - The consumers (workers) extract the data and perform the indicated banking operation, updating the bank's total balance.

Contents

- 1 Introduction
- 2 Lab Description**
- 3 Material
- 4 Delivery

Development process

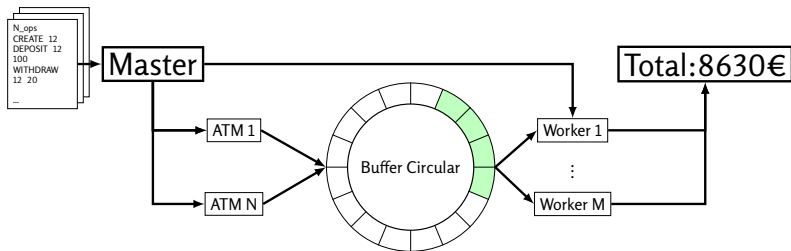


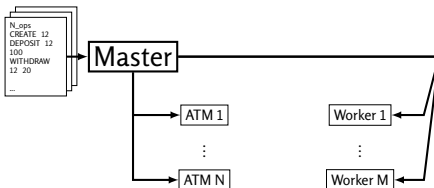
Figure: Example of operation with N ATMs, M workers, and a buffer

Role: Main Process

- In charge of:
 1. Initialize the relevant structures (mutex, condition variables, arrays, ...).
 2. Obtain the operations indicated in the input file and store them in memory (for further processing).
 3. Runs the threads that are part of the Nprod—Mcons system.
 4. Wait for the finish of the threads.
 5. Show the result on screen (*global_balance*).
- Note: It is recommended that you define a structure for each operation and store them in a structure array (AoS).

Role: Producer

1. There can be from 1 up to N producers (value indicated by parameter).
2. Its purpose is to:
 - 1) Safely obtain operation data and increment the client operation variable (*client_numop*).
 - 2) Create an element with the operation data to insert into the shared queue.
 - 3) Insert the element into the shared queue.
 - 4) Once it has finished inserting operations, it ends its execution with *pthread_exit()*.

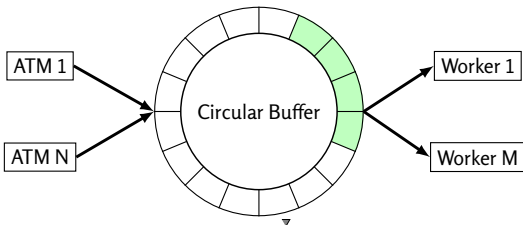


Role: Consumer

- There can be from 1 up to M consumers (value indicated by parameter).
- Its purpose is to:
 1. Increment the bank operation number (*bank_numop*) and extract from the queue the operation with that number.
 2. For each extracted element, perform the indicated bank operation on the associated accounts and also update the global balance.
 3. Print to the screen the type of operation, its parameters, and the resulting balance.
 4. Once it has finished processing the operations, the consumer terminates its execution with *pthread_exit()*.

Circular Buffer

- It is the structure that will store the operations that will be processed.
- It will be used by producers and consumers **concurrently**.
- Access must be concurrent.
- Basic functions defined in *queue.c* & *queue.h*
- Complete operation and queue structure.



Concurrency

- Concurrency control must be implemented using **only mutex & condition variables**.
- It can be done in two situations:
 - *Queue.c*: Concurrency is managed directly in the buffer.
 - *bank.c*: Concurrency is managed between the threads that access the queue.
- Use the method that is easiest for the group
- **Requirement:** The result of the calculation with Nprod- Mcons must be the same as that obtained with 1prod-1cons.

Input

- Input arguments:

`./bank <file> <prods> <cons> <max_accounts> <buff_size>`

- *File*: Input file. It includes the number of operations to be processed and the list of operations with the following format:

```
1 50 #Max number of operations to be processed
2 CREATE 12
3 DEPOSIT 12 100
4 WITHDRAW 12 20
5 CREATE 25
6 TRANSFER 12 25 30
7 BALANCE 25
8 ... # The number of operations must be the number indicated in the first line
```

- *Prods*: Number of producers (ATMs) to be executed
- *Cons*: Number of consumers (workers) to be executed
- *max_accounts*: Maximum number of accounts that the bank can have.
- *buff_size*: Size of the circular buffer (maximum number of elements that can be stored simultaneously).

Output

```
1      $> ./bank input_file 5 3 50 20
2      1 CREATE 12 BALANCE=0 TOTAL=0
3      2 DEPOSIT 12 100 BALANCE=100 TOTAL=100
4      3 WITHDRAW 12 20 BALANCE=80 TOTAL=80
5      4 CREATE 25 BALANCE=0 TOTAL=80
6      5 TRANSFER 12 25 30 BALANCE=30 TOTAL=80
7      6 BALANCE 25 BALANCE=30 TOTAL=80
8      ...
9      $>
```

Contents

- 1 Introduction
- 2 Lab Description
- 3 Material**
- 4 Delivery

Initial Code

- For the development of the programming assignment, an initial code is provided. This code can be downloaded from Aula Global.
- The files given are:

```
1      os_p3_multithread_2023/  
2          Makefile  
3          bank.c  
4          queue.c  
5          queue.h  
6          authors.txt  
7          file.txt  
8          os_checker_p3.sh
```

- To compile, simply execute `make`.
- You must follow the rules included in the statement.

Tester

- An automated tester is provided that offers a tentative grade based on functional tests.
- The automatic tester only runs basic tests. The code will be tested with additional examples for grading.
- Before running:
 - Compress the requested files into a ZIP file with the correct name.
- Execution:

```
./checker_os_p3.sh <zip_file>
```

- Example:

```
$ ./checker_os_p3.sh os_p3_100254896.zip
```


Contents

- 1 Introduction
- 2 Lab Description
- 3 Material
- 4 Delivery**

Assignment Submission & Rules

- Groups: 3 students maximum
- Delivery:
 - Source code in a compressed file
 - Lab report in PDF through TURNITIN
 - Only one member of the group may deliver
- Delivery date:

May 12th 2023 (until 23:55h)



Computer Architecture and Technology Area

Universidad Carlos III de Madrid

OPERATING SYSTEMS

Lab 3. Multithreading

Bachelor's Degree in Computer Science & Engineering
Bachelor's Degree in Applied Mathematics & Computing
Dual Bachelor in Computer Science & Engineering & Business
Administration

Year 2022-2023