



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
Facultad de Ingeniería



Ingeniería en Ciencias de la Computación
COMPUTO PARALELO Y DISTRIBUÍDO
1.23 Actividad 10: Semaphore

Trabajo de: ADRIAN (ADORA) GONZÁLEZ DOMÍNGUEZ [359834]

Asesor: JOSE SAUL DE LIRA MIRAMONTES

17 de septiembre de 2024

```
import logging
import threading

logging.basicConfig(level=logging.DEBUG, format='%(threadName)s:
%(message)s')

# Clase para representar una cuenta bancaria
class BankAccount:
    def __init__(self, initial_balance=0):
        self.balance = initial_balance
        self.semaphore = threading.Semaphore(1) # Semáforo para
        controlar el acceso a la cuenta

    # Método para depositar dinero en la cuenta
    def deposit(self, amount):
        with self.semaphore: # Controla el acceso a la cuenta con un
            semáforo
            self.balance += amount
            logging.info(f'Deposit of {amount}. Current balance:
{self.balance}')

    # Método para retirar dinero de la cuenta
    def withdraw(self, amount):
        with self.semaphore: # Controla el acceso a la cuenta con un
            semáforo
            if self.balance >= amount:
                self.balance -= amount
                logging.info(f'Withdrawal of {amount}. Current
balance: {self.balance}')
            else:
                logging.warning(f'Failed withdrawal. Insufficient
balance: {self.balance}')

# Función para simular múltiples depósitos
def perform_deposits(account, amount, times):
    for _ in range(times):
        account.deposit(amount)

# Función para simular múltiples retiros
```

```
def perform_withdrawals(account, amount, times):
    for _ in range(times):
        account.withdraw(amount)

if __name__ == '__main__':
    # Crear una cuenta bancaria con un saldo inicial de 300
    account = BankAccount(initial_balance=300)

    # Crear hilos para realizar depósitos y retiros
    deposit_thread = threading.Thread(target=perform_deposits,
    args=(account, 100, 100))
    withdrawal_thread = threading.Thread(target=perform_withdrawals,
    args=(account, 50, 200))

    # Iniciar los hilos
    deposit_thread.start()
    withdrawal_thread.start()

    # Esperar a que los hilos terminen
    deposit_thread.join()
    withdrawal_thread.join()

    # Imprimir el saldo final de la cuenta
    logging.info(f'Final account balance: {account.balance}')
```

```
thead@adrigondo MINGW64 ~/Documents/UACH/Seventh Semester/Parallel and Distributing Computing/code
$ python PDC.September\ 17\, \ 2024.Activity\ 1.24.py
```

```
Thread-1 (perform_deposits): Deposit of 100. Current balance: 400
Thread-1 (perform_deposits): Deposit of 100. Current balance: 500
Thread-1 (perform_deposits): Deposit of 100. Current balance: 600
Thread-1 (perform_deposits): Deposit of 100. Current balance: 700
Thread-1 (perform_deposits): Deposit of 100. Current balance: 800
Thread-1 (perform_deposits): Deposit of 100. Current balance: 900
Thread-1 (perform_deposits): Deposit of 100. Current balance: 1000
Thread-1 (perform_deposits): Deposit of 100. Current balance: 1100
Thread-1 (perform_deposits): Deposit of 100. Current balance: 1200
Thread-1 (perform_deposits): Deposit of 100. Current balance: 1300
```

```
Thread-1 (perform_deposits): Deposit of 100. Current balance: 9900
Thread-1 (perform_deposits): Deposit of 100. Current balance: 10000
Thread-1 (perform_deposits): Deposit of 100. Current balance: 10100
Thread-1 (perform_deposits): Deposit of 100. Current balance: 10200
Thread-1 (perform_deposits): Deposit of 100. Current balance: 10300
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 10250
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 10200
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 10150
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 10100
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 10050
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 10000
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 9950
```

```
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 800
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 750
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 700
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 650
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 600
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 550
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 500
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 450
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 400
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 350
Thread-2 (perform_withdrawals): Withdrawal of 50. Current balance: 300
MainThread: Final account balance: 300
```