Fr. Conceicao Rodrigues College of Engineering Department of Computer Engineering			
Student's Roll No	**9913**	**Students Name**	**Mark Lopes**
Date of Performance	**7/03/2024**	**SE Computer – Div**	**A**

Aim: Study Multiprocessing and Process Synchronization

Lab Outcome:

CSL403.3: Understand and apply the concepts of synchronization and deadlocks

Pre-requirement: Python Programming.

Problem Statements:

1. **WAP to demonstrate how to use lock mechanism to achieve process synchronization.**
2. **WAP to demonstrate the use of Queue mechanism to achieve process synchronization in Producer – Consumer Problem.**

The outputs should reflect behaviour of processes with and without process synchronization in both techniques.

References:

https://www.youtube.com/watch?v=RR4SoktDQAw&list=PL5tcWHG-UPH3SX16DI6EP1FlEibgxkg_6&index=1

https://www.youtube.com/watch?v=iYJNmuD4McE&list=PL5tcWHG-UPH3SX16DI6EP1FlEibgxkg_6&index=3

https://www.youtube.com/watch?v=TQx3IfCVvQ0&list=PL5tcWHG-UPH3SX16DI6EP1FlEibgxkg_6&index=6

On time Submission(2)	Knowledge of Topic(4)	Implementation and Demonstraion(4)	Total (10)
Signature of Faculty		**Date of Submission**	

With locks:

```
import time
from multiprocessing import Process, Lock, Value


def deposit_with_lock(balance, lock, amount):
    for i in range(100):
        time.sleep(0.01)
        lock.acquire()
        balance.value += round(amount / 100)
        lock.release()


def withdraw_with_lock(balance, lock, amount):
    for i in range(100):
        time.sleep(0.01)
        lock.acquire()
        balance.value -= round(amount / 100)
        lock.release()


if __name__ == '__main__':
    initial_balance = int(input("Enter initial balance: "))
    balance = Value('i', initial_balance)
    lock = Lock()


    deposit_amount = int(input("Enter the deposit amount: "))
    withdraw_amount = int(input("Enter the withdrawal amount: "))


    # Transaction with locks
    print("\nTransaction with locks:")
```

```
    deposit_proc_with_lock = Process(target=deposit_with_lock,
args=(balance, lock, deposit_amount))

    withdraw_proc_with_lock = Process(target=withdraw_with_lock,
args=(balance, lock, withdraw_amount))

    deposit_proc_with_lock.start()

    withdraw_proc_with_lock.start()

    deposit_proc_with_lock.join()

    withdraw_proc_with_lock.join()



    print("Final balance with locks:", balance.value)
```

```
PS C:\Users\Mark Lopes\Desktop\college\Sem_4\Os\Lab5> python -u "c:\Users\Mark Lopes\Desktop\college\Sem_4\Os\Lab5\lock.py"
Enter initial balance: 1000
Enter the deposit amount: 200
Enter the withdrawal amount: 100

Transaction with locks:
Final balance with locks: 1100
PS C:\Users\Mark Lopes\Desktop\college\Sem_4\Os\Lab5>
```

Without locks:

```
import time

from multiprocessing import Process, Lock, Value



def deposit_without_lock(balance, amount):

    for i in range(100):

        time.sleep(0.01)

        balance.value += round(amount / 100)



def withdraw_without_lock(balance, amount):

    for i in range(100):

        time.sleep(0.01)

        balance.value -= round(amount / 100)



if __name__ == '__main__':
```

```
    initial_balance = int(input("Enter initial balance: "))

    balance = Value('i', initial_balance)

    lock = Lock()



    deposit_amount = int(input("Enter the deposit amount: "))

    withdraw_amount = int(input("Enter the withdrawal amount: "))



    # Transaction without locks

    print("\nTransaction without locks:")

    deposit_proc_without_lock = Process(target=deposit_without_lock,
args=(balance, deposit_amount))

    withdraw_proc_without_lock = Process(target=withdraw_without_lock,
args=(balance, withdraw_amount))

    deposit_proc_without_lock.start()

    withdraw_proc_without_lock.start()

    deposit_proc_without_lock.join()

    withdraw_proc_without_lock.join()



    print("Final balance without locks:", balance.value)
```

```
PS C:\Users\Mark Lopes\Desktop\college\Sem_4\Os\Lab5> python -u "c:\Users\Mark Lopes\Desktop\college\Sem_4\Os\Lab5\nolock.py"
Enter initial balance: 1000
Enter the deposit amount: 200
Enter the withdrawal amount: 100

Transaction without locks:
Final balance without locks: 1060
```