**EX.NO:8**

**DATE:**

**IMPLEMENTATION OF CONCURRENT PROGRAMMING PROGRAM**

**AIM:**

To Write a Python Programs

1. to implement race condition in concurrent program

2. to implement the concurrent program using Locks

**ALGORITHM 1:**

* Start the program
* Import THREADING
* Define SHARED VARIABLE
* Define functions
* Setup Threads and Start it
* Execute the program

**PROGRAM 1:**

import threading

x = 0 # A shared value

COUNT = 10

def incr():

global x

for i in range(COUNT):

x += 1

print(x)

def decr():

global x

for i in range(COUNT):

x -= 1

print(x)

t1 = threading.Thread(target=incr)

t2 = threading.Thread(target=decr)

t1.start()

t2.start()

t1.join()

t2.join()

print(x)

**OUTPUT 1:**

Output first time : Output second time :

|  |  |
| --- | --- |
|  |  |

**ALGORITHM 2:**

* Start the program
* Import THREADING
* Define SHARED VARIABLE
* Define functions using ACQUIRE , RELEASE functions
* Setup Threads and Start it
* Execute the program

**PROGRAM 2:**

import threading

x = 0 # A shared value

COUNT = 10

lock = threading.Lock()

def incr():

global x

lock.acquire()

print("thread locked for increment cur x=",x)

for i in range(COUNT):

x += 1

print(x)

lock.release()

print("thread release from increment cur x=",x)

def decr():

global x

lock.acquire()

print("thread locked for decrement cur x=",x)

for i in range(COUNT):

x -= 1

print(x)

lock.release()

print("thread release from decrement cur x=",x)

t1 = threading.Thread(target=incr)

t2 = threading.Thread(target=decr)

t1.start()

t2.start()

t1.join()

t2.join()

**OUTPUT 2:**

Output count=10: Output count=5 :

|  |  |  |
| --- | --- | --- |
|  |  |  |