NAME : Priyanshu

Roll no : 2023UCS1586

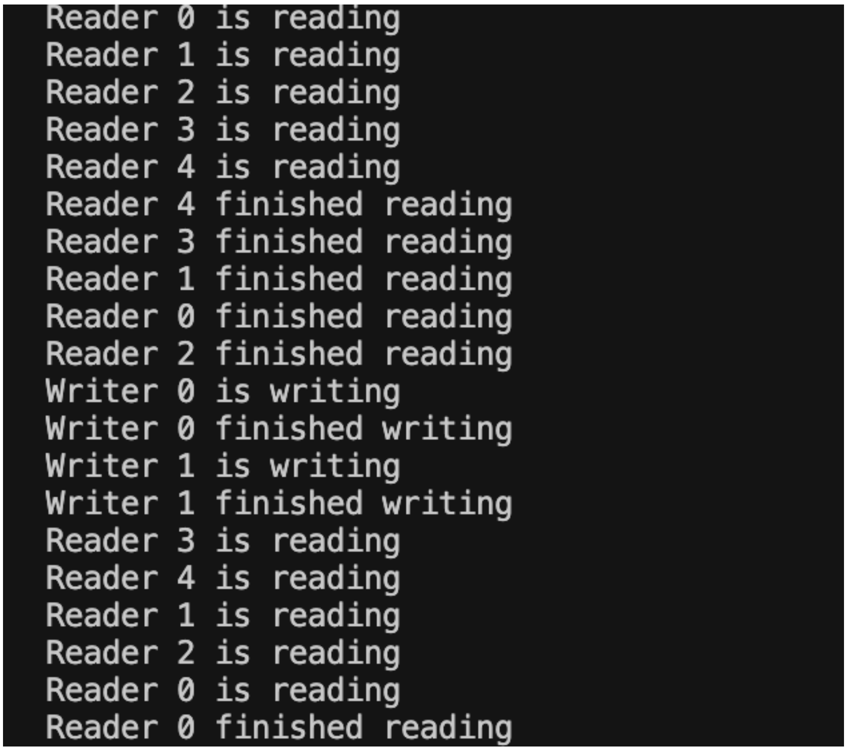
SUBJECT : OS LAB

Question 1) Write a program to implement solution of problem of reader-writer program

code:

```
import threading
import time
import random

class ReaderWriterProblem:
    def __init__(self):
        self.reader_count = 0
        self.reader_count_lock = threading.Lock()
        self.resource_lock = threading.Lock()

    def reader(self, reader_id):
        while True:
            with self.reader_count_lock:
                self.reader_count += 1
                if self.reader_count == 1:
                    self.resource_lock.acquire()

            print(f"Reader {reader_id} is reading")
            time.sleep(random.uniform(0.5, 1.5))
            print(f"Reader {reader_id} finished reading")

            with self.reader_count_lock:
                self.reader_count -= 1
                if self.reader_count == 0:
                    self.resource_lock.release()

            time.sleep(random.uniform(1, 3))

    def writer(self, writer_id):
        while True:
            self.resource_lock.acquire()


            print(f"Writer {writer_id} is writing")
            time.sleep(random.uniform(1, 2))
            print(f"Writer {writer_id} finished writing")

            self.resource_lock.release()
            time.sleep(random.uniform(2, 4))

if __name__ == "__main__":
    problem = ReaderWriterProblem()

    readers = [threading.Thread(target=problem.reader, args=(i,))
for i in range(5)]
    writers = [threading.Thread(target=problem.writer, args=(i,))
for i in range(2)]

    for reader in readers:
        reader.start()

    for writer in writers:
        writer.start()

    for reader in readers:
        reader.join()

    for writer in writers:
        writer.join()
```

Output :



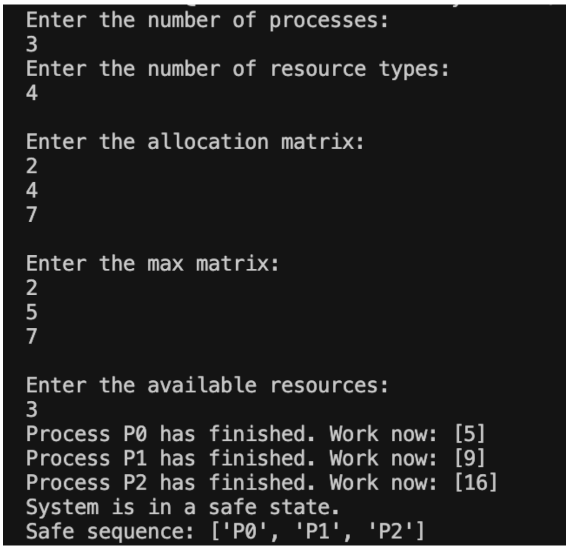
```
Reader 0 is reading
Reader 1 is reading
Reader 2 is reading
Reader 3 is reading
Reader 4 is reading
Reader 4 finished reading
Reader 3 finished reading
Reader 1 finished reading
Reader 0 finished reading
Reader 2 finished reading
Writer 0 is writing
Writer 0 finished writing
Writer 1 is writing
Writer 1 finished writing
Reader 3 is reading
Reader 4 is reading
Reader 1 is reading
Reader 2 is reading
Reader 0 is reading
Reader 0 finished reading
```

Question 2) Write a Program to implement bankers algorithm. this program should be generalised for any number of processes and it should be generic.

Code :

```
def is_safe(processes, avail, max_matrix, alloc):
    num_processes = len(processes)
    num_resources = len(avail)

    need = [[max_matrix[i][j] - alloc[i][j] for j in 
range(num_resources)]
            for i in range(num_processes)]

    work = avail[:]
    finish = [False] * num_processes

    safe_sequence = []

    while len(safe_sequence) < num_processes:
        allocated_in_round = False

        for i in range(num_processes):

            if not finish[i] and all(need[i][j] <= 
work[j] for j in range(num_resources)):
                for j in range(num_resources):
                    work[j] += alloc[i][j]

                finish[i] = True
                safe_sequence.append(processes[i])
                allocated_in_round = True
                print(f"Process {processes[i]} has 
finished. Work now: {work}")
                break

        if not allocated_in_round:
            print("System is in an unsafe state. No 
safe sequence exists.")
            return False
    print("System is in a safe state.")
    print(f"Safe sequence: {safe_sequence}")
    return True

def main():
    print("Enter the number of processes: ")
    num_processes = int(input())
    print("Enter the number of resource types: ")
    num_resources = int(input())

    processes = [f"P{i}" for i in
range(num_processes)]

    print("\nEnter the allocation matrix:")
    alloc = [[int(x) for x in input().split()] for _
in range(num_processes)]

    print("\nEnter the max matrix:")
    max_matrix = [[int(x) for x in input().split()]
for _ in range(num_processes)]

    print("\nEnter the available resources:")
    avail = [int(x) for x in input().split()]

    is_safe(processes, avail, max_matrix, alloc)

if __name__ == "__main__":
    main()
```

Output :



```
Enter the number of processes:
3
Enter the number of resource types:
4

Enter the allocation matrix:
2
4
7

Enter the max matrix:
2
5
7

Enter the available resources:
3
Process P0 has finished. Work now: [5]
Process P1 has finished. Work now: [9]
Process P2 has finished. Work now: [16]
System is in a safe state.
Safe sequence: ['P0', 'P1', 'P2']
```