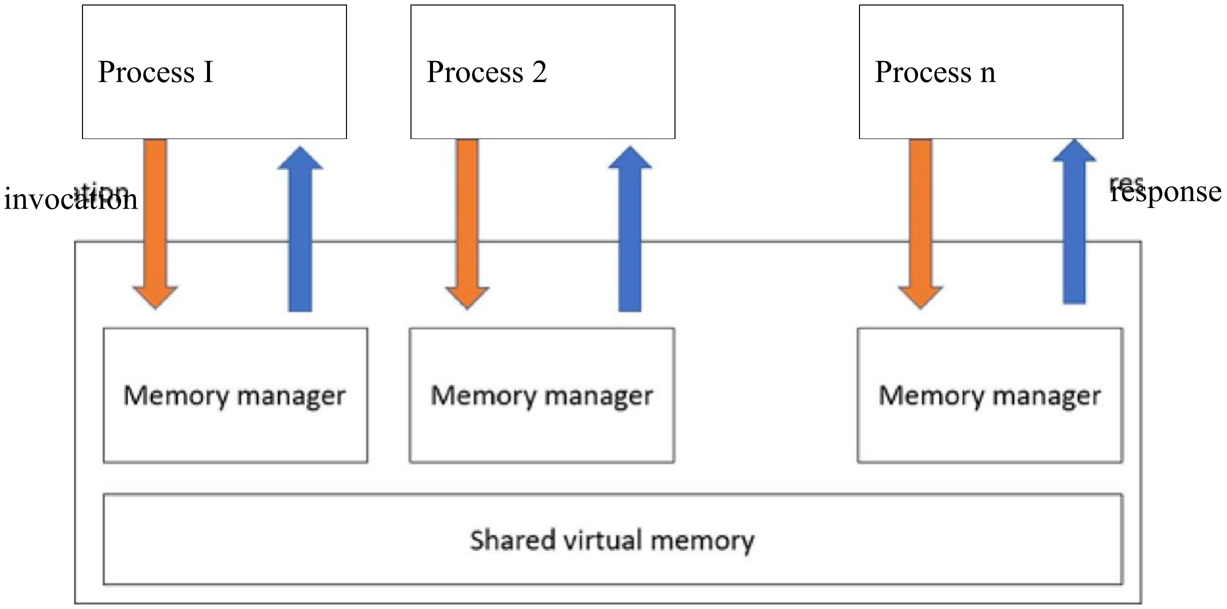
EXPERIMENT NO : 10

AIM: Distributed Shared Memory.

Theory:

Distributed Shared Memory :

DSM is a mechanism that manages memory across multiple nodes and makes inter-process communications transparent to end-users. The applications will think that they are running on shared memory. DSM is a mechanism of allowing user processes to access shared data without using interprocess communications. In DSM every node has its own memory and provides memory read and write services and it provides consistency protocols. The distributed shared memory (DSM) implements the shared memory model in distributed systems but it does not have physical shared memory. All the nodes share the virtual address space provided by the shared memory model. The Data moves between the main memories of different nodes.



Types of Distributed shared memory :

On-Chip Memory:

- The data is present in the CPU portion of the chip.
- Memory is directly connected to address lines.
- On-Chip Memory DSM is expensive and complex.

Bus-Based Multiprocessors:

- A set of parallel wires called a bus act as a connection between CPU and memory.
- Accessing of same memory simultaneously by multiple CPUs is prevented by using some algorithms
- Cache memory is used to reduce network traffic.

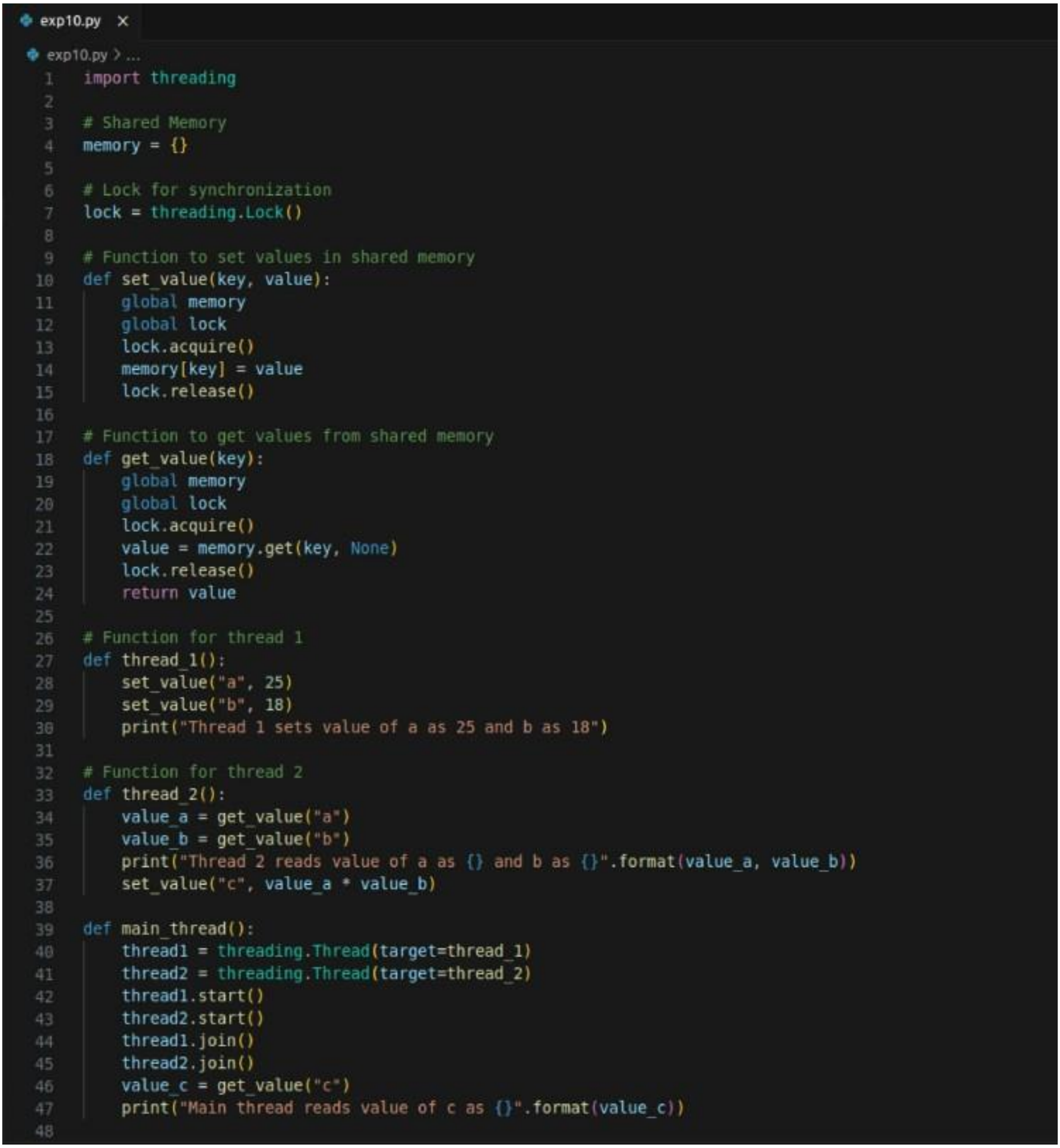
Ring-Based Multiprocessors:

- There is no global centralized memory present in Ring-based DSM.
- All nodes are connected via a token passing ring.
- In ring-bases DSM a single address line is divided into the shared area.

Advantages of Distributed shared memory :

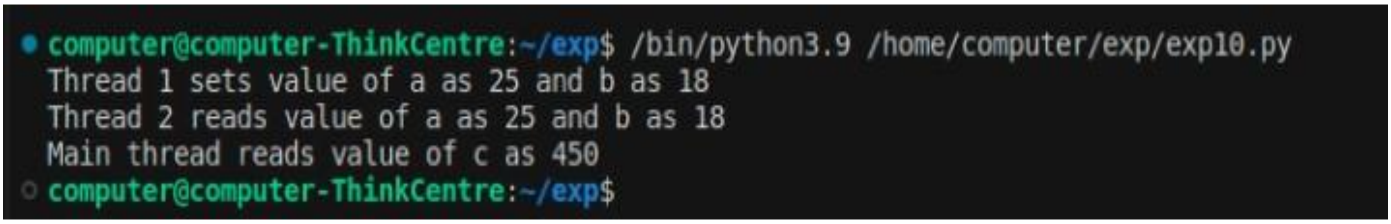
- Simpler abstraction: Programmer need not concern about data movement, As the address space is the same it is easier to implement than RPC.
- Easier portability: The access protocols used in DSM allow for a natural transition from sequential to distributed systems. DSM programs are portable as they use a common programming interface.
- locality of data: Data moved in large blocks i.e. data near to the current memory location that is being fetched, may be needed future so it will be also fetched.
- on-demand data movement: It provided by DSM will eliminate the data exchange phase.
- larger memory space: It provides large virtual memory space, the total memory size is the sum of the memory size of all the nodes, paging activities are reduced.
- Better Performance: DSM improve performance and efficiency by speeding up access to data.
- Flexible communication environment: They can join and leave DSM system without affecting the others as there is no need for sender and receiver to existing, • process migration simplified: They all share the address space so one process can easily be moved to a different machine.

CODE :



```python
import threading

# Shared Memory
memory = {}

# Lock for synchronization
lock = threading.Lock()

# Function to set values in shared memory
def set_value(key, value):
    global memory
    global lock
    lock.acquire()
    memory[key] = value
    lock.release()

# Function to get values from shared memory
def get_value(key):
    global memory
    global lock
    lock.acquire()
    value = memory.get(key, None)
    lock.release()
    return value

# Function for thread 1
def thread_1():
    set_value("a", 25)
    set_value("b", 18)
    print("Thread 1 sets value of a as 25 and b as 18")

# Function for thread 2
def thread_2():
    value_a = get_value("a")
    value_b = get_value("b")
    print("Thread 2 reads value of a as {} and b as {}".format(value_a, value_b))
    set_value("c", value_a * value_b)

def main_thread():
    thread1 = threading.Thread(target=thread_1)
    thread2 = threading.Thread(target=thread_2)
    thread1.start()
    thread2.start()
    thread1.join()
    thread2.join()
    value_c = get_value("c")
    print("Main thread reads value of c as {}".format(value_c))

```

Output :



```
computer@computer-ThinkCentre:~/exp$ /bin/python3.9 /home/computer/exp/exp10.py
Thread 1 sets value of a as 25 and b as 18
Thread 2 reads value of a as 25 and b as 18
Main thread reads value of c as 450
computer@computer-ThinkCentre:~/exp$
```

Conclusion:

Thus, we had Successfully Implemented Distributed shared memory.