Ex. No.: 6

Date: 19.3.24

IPC USING SHARED MEMORY

Aim:

To write a C program to do Inter Process Communication (IPC) using shared memory between sender process and receiver process.

Algorithm:

SENDER

1. Set the size of the shared memory segment
2. Allocate the shared memory segment using shmget
3. Attach the shared memory segment using shmat
4. Write a string to the shared memory segment using sprintf
5. Set delay using sleep
6. Detach shared memory segment using shmdt

RECEIVER

1. Set the size of the shared memory segment
2. Allocate the shared memory segment using shmget
3. Attach the shared memory segment using shmat
4. Print the shared memory contents sent by the sender process.
5. Detach shared memory segment using shmdt

Program Code:

```
# writer.py
import os
import sys
pipe-name = "/home/osuser331/rec.fifo"
if not os.path.exists(pipe_name):
        os.mkfifo(pipe-name)
fifo = open(pipe-name, 'w')
mystring= input(" Enter string to write into pipe: ")
fifo.write (mystring)
fifo.close ()
```

```
# reader.py
    import os
    import sys
    pipe_name = "/home/osuser331/rec.fifo"
    fifo = open(pipe_name, 'r')
    stringRecieved = fifo.read()
    print("string Recieved: ", stringRecieved)
    fifo.close()
```

Output:

```
python3 writer.py
Enter string to write into pipe:
                    Hello, world!
```

```
python3 reader.py
String Recieved :  Hello, world!
```

RESULT:

The programs have been executed and the output has been verified successfully.

19/3/24

10/10