Ex. No.: 7

Date: 28/3/25

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
- 5. Detach shared memory segment using shmdt

Program Code:

sender.c

include <stdio. ht 4 Parlua Estalib. hy

indude Lsyslipe. L7

include <sys/3hm.h7

include / unista.h7
include / string.h7
define SHM_SIZE 1024

```
int main() &
 key-t . key = ftok (" Shm file", 65);
 "In show = shought ( ky, SHUSIZE, 0666 TT( = cueste);
    if (shmia == -1) {
            Parmor ( " Shonget failed ");
           mit(i);
    4
   Char + Shmaddr = (char +) Shmat (shmid, NOLL = 0);
   if LShmaddr == Lchar + ) -1) {
          Person ("Shout failed");
          enit (1);
    printf (shmaddr, "Hello from suder process!");
    prints ("Sindu: Data written be shown memory: 1/5 ln",
                                                shmaddy);
     stup (10);
     shindt. (shinodd r);
      return o;
   4
```

AAAA

```
receiver.c
# include & Stolio. h7
# include establis. h7
# Pached Lsys lipe . hy
# include & systohm.h >
# include & unistd. h7
# define SHM. SIZE LO24
 in+ main() &
       key-t ky = ftok ("Shon fill", 65);
      int should = Shonget Clay, SHN - SIZE, 0666);
      if ( Showid = = -1) {
               Peroson ("Shinget failed");
               enit (1);
      z
    (har + Shmaddr = (char +) Shm at (shmid, NULL, 0);
     if (Shmadh = = (char + ) - 1) {
           peros ("Shmar failed");
            onit (1);
  print f l'Recissur: Dataread from Shared mimory: 1.5 ln"
                                                 shmaddr);
   should (Should');
   should ( Simil , IPL , RMID , NULL);
   return o;
3
```

Sample Output

Terminal 1

[root@localhost student]# gcc sender.c -o sender [root@localhost student]# ./sender

Terminal 2

[root@localhost student]# gcc receiver.c -o receiver [root@localhost student]# ./receiver Message Received: Welcome to Shared Memory [root@localhost student]#

output 1:

Sinder: Data writter to shared memory! Hello from Sinder process!

output 2:

Recieve : Data mad from Shared memory : Hello from sunder proces!

Result:

Thus, the program for interprous communication between another & necience is executed successfully.