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:

### <u>sender</u>

- 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 < state. h >
# include < stallib. h>
# include < sys/ipc. h>
# include < orys/oshm.h>
# conclude <unestal. h >
# include < string. h
# define SIZE 1024
int raciod) &
       koyet key=1234;
        int societ;
        Chan & glowed men;
         Thought = stronget (key, SIZE, O666 IPC_CREAT);
         if ( should = = -1) &
                 person (" shonget failed");
                 exet (1);
         ghowed_mem = (chor *) shmgt(shmid, NULL, 0);
          If Chared_mem == (chan + )-1) $
                  Paroror ("Amost failed");
exit(1);
          special (should rown, "Hellofrom sender Broces!")
          pount ("Sender wrote: "/m", shared_mem).
           glace (5);
          should (shoved-mem);
           neterin 0;
```

0)

0

```
receiver.c
```

```
# include < etotio. h>
# include Zatalla.n>
# include (sys/ipe.h)
# conclude ( sys/ show. h >
# include <unistal. h>
# include < string. h?
# Offine SIZE 1024
Ent main () &
       bey-t key = 1234;
cnt showed;
       Cheer * I drawed - mem;
        shooid = shinget (Dey , SIZE, 0666);
if (shined ==-1) }
                 Peroror (" short failed");
         showed nem = (choor *) showed (showed, NULL, 0);
         If (should-mem == (choor & )-1) &
                  perocor ("shmat faeld");
                    exit (1);
          point ("Receiver seed: %2\n" Shoved mem);
          shmot ( showed mem);
          shmeth (shined, IR-RMID, NULL);
          getwen 0;
```

# 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]#

Tenninal 1

\$ . [ sonder · Sender wrote: Hello from Sender Process!

Terrenal 2

& lacines · Receiver read: Hello feron Sendor Porocess!

#### Result:

The Eponogram to do IPC wany schored manage between sender and successfully, eighternated successfully,