Ex. No.: 6

Date: 16.3.23

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:

```
#include <signal.h>
#include <stdio.h>
    void my_handler (int sig);
        int main()
{
    struct sigaction my_action;
```

```
my_action.sa_handler = my_handler;
my_action.sa_flags = SA_Restart;
    Sigaction (SIGINT, &my_action, NULL);
     printf ("Catching SIGINT\n");
      sleep(3);
      printf(" NO SIGINT within 3 seconds");

    /* Part III */

    my_action.sa_handler = SIG_DFL;
    my_action.sa_flags = SA_RESTART;
      Sigaction (SIGINT, &my_action, NULL),
       sleep(3);
       printf("NO SIGINT within 3 seconds");

}
      void my_handler (int sig)
     {
       printf ("It I got SIGINT, number %d
                \n", sig);
          exit(0);
     }
```

Output:

```
Catching SIGINT
NO SIGINT within 3 seconds
Ignoring SIGINT
  sleep is over
NO SIGINT within 3 seconds
```

Result: The above commands are executed successfully

16/3/24

10/10