Ex. No.: 7

Date: 28·03·25

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 process.
5. Detach shared memory segment using shmdt

Program Code:

sender.c

code :

```
#include <stdio.h>
# include <stdlib.h>
# include <sys/ipc.h>
# include <string.h>
# include <uninstd.h>

# define SHM-SIZE 1024
  void sender();
   void receiver();

  ~~void sender();~~
  ~~void receiver();~~

  void sender()
      {
      key-t key = ftok ("shmfile".65);
      int shmid = shmget (key , SHM-SIZE | 0666 |
          IPL-CREAT);
        char* shm-ptr = (char*) shmt (shmid, NULL-0).

        printf (" Enter message : ");
        get char();
        fgets (shm_ptr , SHM-SIZE , stdin);
        print f ("Message sent !\n");
        sleep (10);
        shmdt ( shm.ptr)
      }
```

receiver.c

```
void receiver ()
 {
   key-t key = ftok("Shmfile", 65);
   int shmid = shmget(key, SHM-SIZE, 0666);
   char *shm_ptr = (char *) shmat(shmid, NULL, 0
   printf("Received: %s", shm_ptr);
   shmdt (shm_ptr);
   shmctl(shmid, IPC-RMID, NULL);
   }

  int main(){
    int choice;
    while(1){
      printf("\n1 sender\n2. Receiver \n3.Exit\n
               Enter choice:");
      scanf("%d", &choice);
       {
       case 1:
          sender();
          break;
       case 2:
         receiver ()
          break;
       case 3:
         printf("Exiting...\n");
         exit(0);
      default:
        printf("Invalid choice!");
       }
    }
    return 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]#
```

output:

```
1. Sender
2. Receiver
3. Exit
Enter choice : 1
   Enter message : Hello, receiver!
   Message sent!
     1. Sender
     2. Receiver
     3. Exit
     Enter choice : 2
     Received : Hello, receiver!
```

Result:

Hence the c program to do IPC using shared memory between sender and receiver has been sucessfully executed.