Ex. No.: 7

Date: 02·04·25

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

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>
#define SHM_SIZE 1024
```

```
int main()
{
    key_t key=1234;
    int shmid;
    Char *shm;
    shmid = shmget (key, SHM_SIZE ,0666|IPC_CREAT);
    if( shmid <0){
        perror ("shmget failed");
        return 1;
    }
    shm =(char *) shmat (shmid ,NULL,0);
    if( shm == (char*) -1){
        perror ("shmat failed");
        return1;
    }
    sprintf( shm, "Hello World");
    printf("Recieved message : %s\n", shm);
    shmdt(shm);
    return0;
}
```

receiver.c

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

#include < string.h>
#define SHM_SIZE 1024
```

```
int main() {
    key_t key=1234;
     int shmid;
     char * shm;
    shmid = shmget (key, SHM_SIZE, 0666);
    if(shmid ==-1){
          perror("shmget failed");
          return1;
     }

     shm = (char*) shmat (shmid, NULL, 0);
     if( shm == (char*) -1){
          perror("shmat failed");
          return 1;
     }
     printf("Received message: %s\n", shm);
     shmat(shm);
     return 0;
 }
```

Output:

```
gcc -o senderipc.c
gcc -o receiver ipcl.c

./sender
Received message: Hello World
./reciver
Received message: Hello world
```

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]#
```

Verified

Result: A C program is executed for interprocess communication using shared memory.