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 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 <unistd.h>
#include <string.h>
```

```
#define SIZE 1024

int main(){
    key_t key = 1234;
    int shmid;
    char *shared_mem;
    shmid = shmget(key, SIZE, 0666 | IPC_CREAT);
    if (shmid == -1){
        perror("shmget failed");
        exit(1);
    }
    shared_mem = (char*) shmat(shmid, NULL, 0);
    if (shared_mem == (char*) -1){
        perror("shmat failed");
        exit(1);
    }
    Sprintf(shared_mem, "Hello from Sender Process!");
    printf("Sender wrote: %s\n", shared_mem);
    sleep(5);
    shmat(shared_mem);
    return 0;
}
```

receiver.c

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

#define SIZE 1024

int main(){
    key_t key = 1234;
    int shmid;
    char *shared_mem;
    shmid = shmget(key, SIZE, 0666);
    if (shmid == -1){
        perror("shmget failed");
        exit(1);
    }
    shared_mem = (char*) shmat(shmid, NULL, 0);
    if (shared_mem == (char*) -1){
        perror("shmat failed");
        exit(1);
    }
    printf("Reciever read : %s\n", shared_mem);
    shmdt(shared_mem);
    shmctl(shmid, IPC_RMID, NULL);
    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]#
```

Result:

Program to do Interprocess Communication (IPC) using shared memory between sender process & receiver process was compiled and executed successfully.

Output:

Sender wrote: Hello from Sender Process!.

Reciever read: Hello from Sender Process!.