Ex. No.: 7

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.

Program Code:

Sender.c

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

int main() {
    key_t key = ftok("shmfile", 65);  // generate unique key
    int shmid = shmget(key, 1024, 0666|IPC_CREAT); // create shared memory

    char *str = (char*) shmat(shmid, (void*)0, 0); // attach to shared memory

    sprintf(str, "Welcome to Shared Memory"); // write to shared memory
    printf("Message Sent: %s\n", str);

    sleep(5); // delay to allow receiver to read

    shmdt(str); // detach from shared memory

    return 0;
}
```

Receiver.c

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

int main() {
    key_t key = ftok("shmfile", 65); // same key as sender
    int shmid = shmget(key, 1024, 0666|IPC_CREAT); // get shared memory

    char *str = (char*) shmat(shmid, (void*)0, 0); // attach to shared memory
```

```
    printf("Message Received: %s\n", str); // read from shared memory

    shmdt(str); // detach from shared memory

    shmctl(shmid, IPC_RMID, NULL); // optional: remove shared memory

    return 0;
}
```

OUTPUT:

```
[root@localhost student]# gcc sender.c -o sender
[root@localhost student]# ./sender
Message Sent: Welcome to Shared Memory
```

```
[root@localhost student]# gcc receiver.c -o receiver
[root@localhost student]# ./receiver
Message Received: Welcome to Shared Memory
```