Ex. No: 7
Date: 22/2/25

IPC USING SHARED MEMORY

AIM:

To write a C program to do Inter-Process Communication (IPC) using shared memory between the sender process and the 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:
SENDER

```
int main() {
    key_t key = ftok("sender.c", 65);
    int shmid;
    SharedMemory *shm;

    shmid = shmget(key, sizeof(SharedMemory), 0666 | IPC_CREAT);
    if (shmid == -1) {
        perror("shmget failed");
        exit(1);
    }

    shm = (SharedMemory *)shmat(shmid, NULL, 0);
    if (shm == (SharedMemory *)-1) {
        perror("shmat failed");
        exit(1);
    }

    printf("Sender: Enter a message to send to receiver:
    "); fgets(shm->message, SHMSIZE, stdin);

    shm->message[strcspn(shm->message, "\n")] = '\0';


    shm->ready = 1;

    sleep(5);

    if (shmdt(shm) == -1) {
        perror("shmdt failed");
        exit(1);
    }

    return 0;
}
```

RECEIVER

```
int main() {
    key_t key = ftok("sender.c", 65);
    int shmid;
    SharedMemory *shm;

    shmid = shmget(key, sizeof(SharedMemory), 0666 | IPC_CREAT);
    if (shmid == -1) {
        perror("shmget failed");
        exit(1);
    }

    shm = (SharedMemory *)shmat(shmid, NULL, 0);
    if (shm == (SharedMemory *)-1) {
        perror("shmat failed");
        exit(1);
    }


    while (shm->ready == 0) {
        sleep(1);
    }
    printf("Receiver: Message received from sender: %s\n", shm->message);

    if (shmdt(shm) == -1) {

        perror("shmdt failed");
        exit(1);
    }

    if (shmctl(shmid, IPC_RMID, NULL) == -1) {
        perror("shmctl failed");
        exit(1);
    }

    return 0;
}
```

OUTPUT: