Name	SHIVSHARAN GURUNATH SANJAWAD
UID no.	2023300194
Experiment No.	8

AIM:	IPC using shared memory

Reader :
```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <semaphore.h>
#include <string.h>
#include <unistd.h>
#include "shared.h"

int main() {
        key_t key = ftok("shmfile", 35);
        int shmid = shmget(key, sizeof(struct shmseg), 0666 | IPC_CREAT);
        struct shmseg *shm = (struct shmseg *)shmat(shmid, NULL, 0);
        while (shm->initialized != 1) {
        usleep(100000);
        }

        for (int i = 0; i < 2; i++) {
        sem_wait(&shm->sem_reader);
        printf("Reader got: %s\n", shm->data);
        sem_post(&shm->sem_writer);
        sleep(2);
        }

        shmdt(shm);
        return 0;
}
```

Writter :
```
#include <stdio.h>
#include <sys/ipc.h>
```

```
#include <sys/shm.h>
#include <semaphore.h>
#include <string.h>
#include <unistd.h>
#include "shared.h"

int main() {
        key_t key = ftok("shmfile", 35);
        int shmid = shmget(key, sizeof(struct shmseg), 0666 | IPC_CREAT);
        struct shmseg *shm = (struct shmseg *)shmat(shmid, NULL, 0);

        if (shm->initialized != 1) {
        sem_init(&shm->sem_writer, 1, 1);
        sem_init(&shm->sem_reader, 1, 0);
        shm->initialized = 1;
        }

        for (int i = 0; i < 2; i++) {
        sem_wait(&shm->sem_writer);
        if (i == 0)
        strcpy(shm->data, "Hello from Writter");
        else
        strcpy(shm->data, "Again Hello from Writter");
        sem_post(&shm->sem_reader);
        sleep(2);
        }

        shmdt(shm);
        return 0;
}

Shared
#ifndef SHARED_H
#define SHARED_H

#include <semaphore.h>

struct shmseg {
        int initialized;
        sem_t sem_writer;
        sem_t sem_reader;
        char data[1024];
};
```
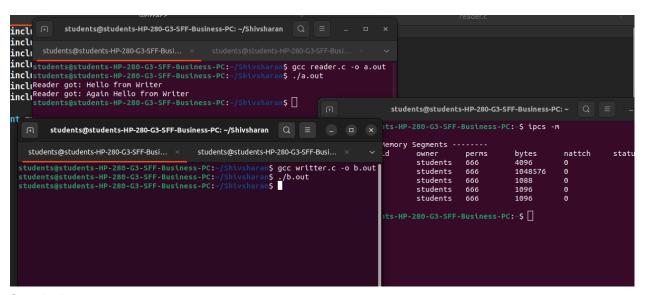
#endif

Conclusion :

Learnt how to use the shared memory between two different processes and how to check existing shared memories and also communicate between processes.