**//SERVER SIDE**

#include<stdio.h>

#include<sys/ipc.h>

#include<sys/shm.h>

struct dbase{

char buffer[1024];

int status;

};

int main()

{

key\_t key = ftok("shmfile",65);

int shmid = shmget( key, sizeof(struct dbase), 0666 | IPC\_CREAT);

struct dbase \*str = (struct dbase\*) shmat( shmid, (void\*) 0, 0);

str->status =0;

printf("\nWrite the data :");

gets(str->buffer);

str->status=1;

while(str->status !=0){

printf("\nWaiting");

sleep(1);

}

printf("\nData written in the memory : %s\n", str->buffer );

shmdt(str);

return 0;

}

**//CLIENT SIDE**

#include<stdio.h>

#include<sys/ipc.h>

#include<sys/shm.h>

struct dbase{

char buffer[1024];

int status;

};

int main()

{

key\_t key = ftok("shmfile",65);

int shmid = shmget( key,0, 0666);

struct dbase \*str = (struct dbase \*) shmat( shmid,(void\*) 0, 0);

while(str->status!=1){

printf("\nWaiting");

sleep(1);

}

printf("\nData read from the memory : %s\n", str->buffer);

str->status=1;

printf("\nWrite the data :");

gets(str->buffer);

str->status=0;

shmdt(str);

shmctl( shmid, IPC\_RMID, NULL ); //destroy the shared memory

return 0;

}

**// OUTPUT**

[student@localhost ~]$ gcc a.c

[student@localhost ~]$ ./a.out

Write the data :os practical sessions

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Waiting

Data written in the memory : done program

[student@localhost ~]$ gcc b.c -o b.out

[student@localhost ~]$ ./b.out

Data read from the memory : os practical sessions

Write the data :done program