LAB 3: INTERPROCESS COMMUNICATION (SHARED MEMORY)

SOUCE CODE:

PROGRAM 1:

#include<stdio.h>

#include<sys/shm.h>

#include<sys/ipc.h>

#include <stdlib.h>

#include <string.h>

int main()

{

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

int shmid=shmget(key ,1024,0666 | IPC\_CREAT);

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

printf("Write data:");

fgets(str,1024,stdin);

printf("Data written in memory:%s\n",str);

return 0 ;

}

PROGRAM 2:

#include <stdio.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdlib.h>

int main()

{

// ftok to generate unique key, same as used by the writer program

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

// shmget returns an identifier in shmid, same as the writer program

int shmid = shmget(key, 1024, 0666 | IPC\_CREAT);

// shmat to attach to shared memory

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

printf("Data read from memory: %s\n", str);

// detach from shared memory

shmdt(str);

// destroy the shared memory

shmctl(shmid, IPC\_RMID, NULL);

return 0;

}

DESCRIPTION:

shmget(): Allocates a shared memory segment. It takes a key, size, and permission flags as arguments and returns an identifier for the shared memory segment.

shmat(): Attaches the shared memory segment to the process's address space. It takes the shared memory identifier, a pointer where the segment should be attached (or NULL for automatic selection), and flags. It returns a pointer to the shared memory segment.

ftok(): Generates a unique key for shared memory or other IPC mechanisms. It takes a pathname and a project identifier as arguments and returns a key based on the provided file and identifier.

OUTPUT:

