Name : Sarthak Pagar

Roll No. : 34

Class : TE (IT)

Practical : 7B

Statement : [Inter-process Communication using Shared Memory using System V. Application to demonstrate: Client and Server Programs in which server process creates a shared memory segment and writes the message to the shared memory segment. Client process reads the message from the shared memory segment and displays it to the screen.](https://drive.google.com/drive/folders/1Q3dOeVAnW_ioSnlMwCAd_d_GcofcTK-w?usp=sharing)

(i) server.c

#include <stdio.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdlib.h>

#include <string.h>

int main() {

// Create a unique key

key\_t key = ftok("shmfile", 65); // Ensure "shmfile" exists in the directory

if (key == -1) {

perror("ftok");

exit(1);

}

// Create a shared memory segment with a size of 1024 bytes

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

if (shmid == -1) {

perror("shmget");

exit(1);

}

// Attach the shared memory segment to the server's address space

char \*str = (char \*) shmat(shmid, NULL, 0); // Corrected type to char\*

if (str == (char \*) -1) {

perror("shmat");

exit(1);

}

printf("Write Data: ");

fgets(str, 100, stdin); // Write data to shared memory

// Detach the shared memory segment

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

perror("shmdt");

exit(1);

}

return 0;

}

(ii) client.c

#include <stdio.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdlib.h>

int main() {

// Create the same key used by the server

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

if (key == -1) {

perror("ftok");

exit(1);

}

// Get the shared memory segment created by the server

int shmid = shmget(key, 1024, 0666);

if (shmid == -1) {

perror("shmget");

exit(1);

}

// Attach the shared memory segment to the client's address space

char \*str = (char \*) shmat(shmid, NULL, 0); // Corrected type to char\*

if (str == (char \*) -1) {

perror("shmat");

exit(1);

}

// Read data from shared memory

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

// Detach the shared memory segment

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

perror("shmdt");

exit(1);

}

// Destroy the shared memory segment

if (shmctl(shmid, IPC\_RMID, NULL) == -1) {

perror("shmctl");

exit(1);

}

return 0;

}

Output :-

sarthak1594@UbuntuInWin:~/Downloads/7B$ gcc -o server server.c

sarthak1594@UbuntuInWin:~/Downloads/7B$ gcc -o client client.c

sarthak1594@UbuntuInWin:~/Downloads/7B$ ./server

Write Data: Hello, My name is Sarthak.

sarthak1594@UbuntuInWin:~/Downloads/7B$ ./client

Data read from memory: Hello, My name is Sarthak.