

Assignment No. 7B

Name : Omkar Hotkar

Roll No : 30

//Server Program

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>

#define SHM_SIZE 1024 // Size of shared memory segment

int main() {
    key_t key;
    int shmid;
    char *data;

    // Generate a unique key
    key = ftok("shmfile", 65);
    if (key == -1) {
        perror("ftok");
        exit(1);
    }

    // Create shared memory segment
    shmid = shmget(key, SHM_SIZE, 0666|IPC_CREAT);
    if (shmid == -1) {
        perror("shmget");
        exit(1);
    }

    // Attach to the shared memory
    data = (char*) shmat(shmid, NULL, 0);
    if (data == (char*)(-1)) {
        perror("shmat");
        exit(1);
    }

    // Write message to shared memory
    printf("Enter a message to send to the client: ");
    fgets(data, SHM_SIZE, stdin);

    printf("Message written to shared memory: %s\n", data);
}
```

```

    // Detach from shared memory
    if (shmdt(data) == -1) {
        perror("shmdt");
        exit(1);
    }

    return 0;
}

```

//Client Program

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>

#define SHM_SIZE 1024 // Size of shared memory segment

int main() {
    key_t key;
    int shmid;
    char *data;

    // Generate the same unique key
    key = ftok("shmfile", 65);
    if (key == -1) {
        perror("ftok");
        exit(1);
    }

    // Locate the shared memory segment
    shmid = shmget(key, SHM_SIZE, 0666);
    if (shmid == -1) {
        perror("shmget");
        exit(1);
    }

    // Attach to the shared memory
    data = (char*) shmat(shmid, NULL, 0);
    if (data == (char*)(-1)) {
        perror("shmat");
        exit(1);
    }

    // Read and display the message from shared memory
    printf("Message from server: %s\n", data);
}

```

```
// Detach from shared memory
if (shmdt(data) == -1) {
    perror("shmdt");
    exit(1);
}

return 0;
}
```

Output:

Enter a message to send to the client: Hello from Server!
Message written to shared memory: Hello from Server!

Message from server: Hello from Server!