

```
//Server.c
```

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

#define SHM_KEY 1234      // Key for shared memory
#define SHM_SIZE 1024     // Size of shared memory

int main() {
    int shmid;
    char *shm_ptr;

    // Create shared memory segment
    shmid = shmget(SHM_KEY, SHM_SIZE, IPC_CREAT | 0666);
    if (shmid < 0) {
        perror("shmget failed");
        exit(EXIT_FAILURE);
    }

    // Attach shared memory to server's address space
    shm_ptr = (char *) shmat(shmid, NULL, 0);
    if (shm_ptr == (char *) -1) {
        perror("shmat failed");
        exit(EXIT_FAILURE);
    }

    // Write message to shared memory
    const char message[] = "Hello Client! Message from Server.";
    strncpy(shm_ptr, message, SHM_SIZE);

    printf("Server: Message written to shared memory.\n");
    printf("Press Enter to detach and exit...\n");
    getchar(); // Wait for user input

    // Detach shared memory
    shmdt(shm_ptr);

    // Remove shared memory segment
    shmctl(shmid, IPC_RMID, NULL);

    return 0;
}
```

```
//Client.c
```

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

#define SHM_KEY 1234
#define SHM_SIZE 1024

int main() {
    int shmid;
    char *shm_ptr;

    // Get the shared memory segment created by server
    shmid = shmget(SHM_KEY, SHM_SIZE, 0666);
    if (shmid < 0) {
        perror("shmget failed");
        exit(EXIT_FAILURE);
    }

    // Attach shared memory to client's address space
    shm_ptr = (char *) shmat(shmid, NULL, 0);
    if (shm_ptr == (char *) -1) {
        perror("shmat failed");
        exit(EXIT_FAILURE);
    }

    // Read message from shared memory
    printf("Client: Message from Server: \"%s\"\n", shm_ptr);

    // Detach shared memory
    shmdt(shm_ptr);

    return 0;
}
```

//OUTPUT

Terminal 1 (Server)

```
manasvi@manasvi:/mnt/c/Assignments/OS/Assignment 7$ gcc client.c -o client
```

```
manasvi@manasvi:/mnt/c/Assignments/OS/Assignment 7$ ./client
```

Client: Message from Server: "Hello Client! Message from Server."

Terminal 2 (Client)

```
manasvi@manasvi:/mnt/c/Assignments/OS/Assignment 7$ gcc client.c -o client
```

```
manasvi@manasvi:/mnt/c/Assignments/OS/Assignment 7$ ./client
```

Client: Message from Server: "Hello Client! Message from Server."

Back to Terminal 1 (Server)After pressing Enter:

Server process terminates.