NAME: ARUN MC

ROLL NO: 230701036

Exp:7 **IPC USING SHARED MEMORY**

Aim:

To write a C program to do Inter Process Communication (IPC) using shared memory between sender process and receiver process.

CODE:

SENDER:

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

#define SHM_SIZE 1024

int main() {
    // Step 1: Generate a unique key
    key_t key = ftok("shmfile", 65);
    // Step 2: Allocate shared memory segment
    int shmid = shmget(key, SHM_SIZE, 0666 | IPC_CREAT);
    // Step 3: Attach shared memory segment
    char *str = (char *) shmat(shmid, NULL, 0);
    // Step 4: Write to shared memory
    sprintf(str, "Hello from Sender!");
    printf("Data written to memory: %s\n", str);
    // Step 5: Simulate delay
    sleep(10);
    // Step 6: Detach from shared memory
    shmdt(str);
    return 0;
}
```

RECEIVER:

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

#define SHM_SIZE 1024

int main() {
    // Step 1: Generate a unique key
    key_t key = ftok("shmfile", 65);
    // Step 2: Allocate shared memory segment
    int shmid = shmget(key, SHM_SIZE, 0666 | IPC_CREAT);
    // Step 3: Attach shared memory segment
    char *str = (char *) shmat(shmid, NULL, 0);
    // Step 4: Print shared memory content
    printf("Data read from memory: %s\n", str);
    printf("using shared memory");
    // Step 5: Detach from shared memory
    shmdt(str);
    // Optional: Destroy the shared memory
    shmctl(shmid, IPC_RMID, NULL);
    return 0;
}
```

OUTPUT:

```
[cse36@localhost ~]$ nano 7_ipc_sender.c
[cse36@localhost ~]$ nano 7_ipc_receiver.c
[cse36@localhost ~]$ gcc 7_ipc_sender.c -o 7_ipc_sender
[cse36@localhost ~]$ gcc 7_ipc_receiver.c -o 7_ipc_receiver
[cse36@localhost ~]$ ./7_ipc_sender &
[1] 4582
[cse36@localhost ~]$ Data written to memory: Hello from Sender!
gcc 7_ipc_sender.c -o./7_ipc_receiver
Data read from memory: Hello from Sender!
using shared memory[cse36@localhost ~]$
```