NAME: G. GANESH

REG NO: 192373008

EXERICSE-9

Illustrate the concept of inter-process communication using shared memory with a C program.

Aim:

To illustrate the concept of inter-process communication (IPC) using shared memory with a C program.

Algorithm:

1. Create a shared memory segment using the shmget system call.
2. Attach the shared memory segment to the process's address space using the shmat system call.
3. In the producer process:
    - Write data to the shared memory.
    - Notify the consumer process of data availability.
4. In the consumer process:
    - Read data from the shared memory.
    - Detach the shared memory segment from the address space using shmdt.
5. Delete the shared memory segment using the shmctl system call.

Procedure:

1. Create two programs: one for the producer and one for the consumer.
2. Use shared memory for data exchange between the producer and consumer.
3. Implement synchronization to ensure proper data exchange.
4. Compile and run both programs to observe IPC.

Code:

```
#include <stdio.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <string.h>

#define SHM_KEY 12345

int main() {
```

```
    int shmid = shmget(SHM_KEY, 1024, 0666 | IPC_CREAT);
    if (shmid == -1) {
        perror("Shared memory");
        return 1;
    }
    char *shared_memory = (char *)shmat(shmid, NULL, 0);
    if (shared_memory == (char *)-1) {
        perror("Shared memory attach");
        return 1;
    }
    printf("Enter a message to write to shared memory: ");
    fgets(shared_memory, 1024, stdin);
    printf("Message written to shared memory: %s\n", shared_memory);
    shmdt(shared_memory);
    return 0;
}
```

Result:

The concept of inter-process communication using shared memory was successfully demonstrated. Data was written by the producer process and read by the consumer process through shared memory.

Output: