

9 .Illustrate the concept of inter-process communication using sharedmemory with a C program.

Program:

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

#define SHM_SIZE 1024

int main() {
    const char *filename = "/tmp/shmfile"; // Use a directory with write
permissions
    int fd = open(filename, O_CREAT | O_RDWR, 0666);
    if (fd == -1) {
        perror("open");
        exit(EXIT_FAILURE);
    }
    close(fd);

    key_t key = ftok(filename, 65);
    if (key == -1) {
        perror("ftok");
        exit(EXIT_FAILURE);
    }

    int shmid = shmget(key, SHM_SIZE, IPC_CREAT | 0666);
    if (shmid == -1) {
        perror("shmget");
        exit(EXIT_FAILURE);
    }

    char *shm_ptr = (char *)shmat(shmid, NULL, 0);
    if (shm_ptr == (char *)(-1)) {
        perror("shmat");
        exit(EXIT_FAILURE);
    }

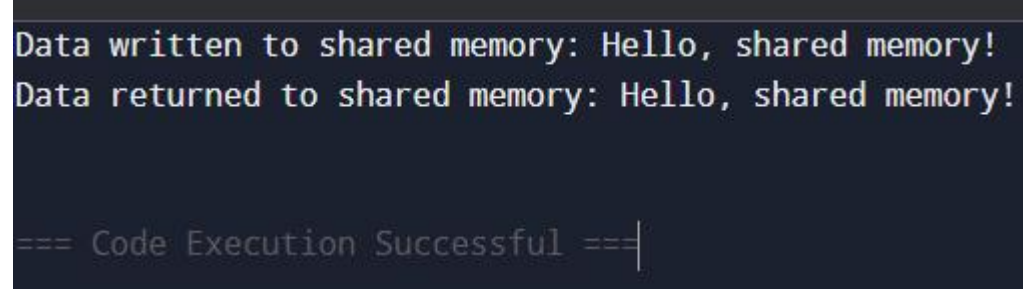
    strcpy(shm_ptr, "Hello, shared memory!");
    printf("Data written to shared memory: %s\n", shm_ptr);

    printf("Data returned to shared memory: %s\n", shm_ptr);

    if (shmdt(shm_ptr) == -1) {
        perror("shmdt");
        exit(EXIT_FAILURE);
    }
}
```

```
    if (shmctl(shmid, IPC_RMID, NULL) == -1) {  
        perror("shmctl");  
        exit(EXIT_FAILURE);  
    }  
  
    return 0;  
}
```

Output:

A screenshot of a terminal window with a dark background. It displays two lines of text: "Data written to shared memory: Hello, shared memory!" followed by "Data returned to shared memory: Hello, shared memory!".

```
Data written to shared memory: Hello, shared memory!  
Data returned to shared memory: Hello, shared memory!
```

```
=== Code Execution Successful ===|
```