THENMOZHI R 231901064

DATE:19.02.2025
EX.NO: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.

Algorithm:

sender

1. Set the size of the shared memory segment
2. Allocate the shared memory segment using shmget
3. Attach the shared memory segment using shmat
4. Write a string to the shared memory segment using sprintf
5. Set delay using sleep
6. Detach shared memory segment using shmdt

receiver

1. Set the size of the shared memory segment
2. Allocate the shared memory segment using shmget
3. Attach the shared memory segment using shmat
4. Print the shared memory contents sent by the sender process.
5. Detach shared memory segment using shmdt

Program Code: sender.c

```
#include <stdio.h> #include <stdlib.h> #include <string.h> #include <sys/shm.h> #include <sys/types.h> #include <unistd.h> #define SHM_SIZE 1024 // Size of shared memory int main() {
```

```
    // Step 1: Set the size of the shared memory segment
    int shmid;
    char *shm_ptr;
    key_t key = 1234; // Shared memory key
```

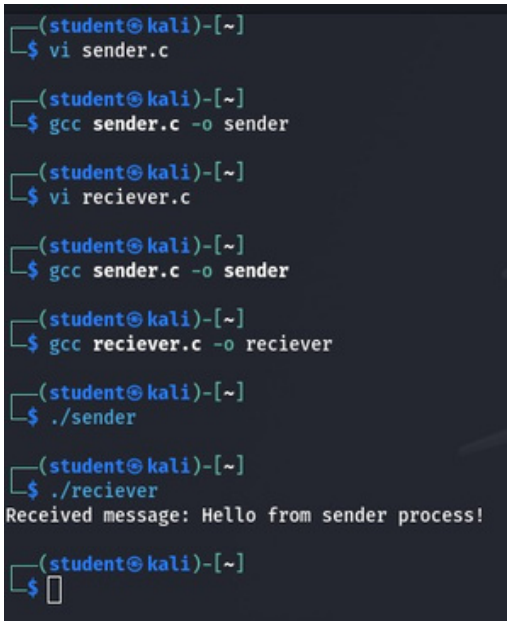
```
    // Step 2: Allocate the shared memory segment
    shmid = shmget(key, SHM_SIZE, 0666 | IPC_CREAT);
    if (shmid == -1) {
    perror("shmget failed");
    exit(1);
    }
    // Step 3: Attach the shared memory segment
    shm_ptr = (char *)shmat(shmid, NULL, 0);
    if (shm_ptr == (char *)(-1)) {
    perror("shmat failed");
    exit(1);
    }
    // Step 4: Write a string to the shared memory segment
    sprintf(shm_ptr, "Hello from sender process!");
    // Step 5: Set delay using sleep (simulating time delay for receiver)
    sleep(5);
    // Step 6: Detach shared memory segment
    if (shmdt(shm_ptr) == -1) {
    perror("shmdt failed");
    exit(1);
    }
    return 0;
}
```

receiver.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/shm.h>
#include <sys/types.h>
#include <unistd.h>
#define SHM_SIZE 1024 // Size of shared memory
int main() {
    // Step 1: Set the size of the shared memory segment
    int shmid;
    char *shm_ptr;
    key_t key = 1234; // Shared memory key
    // Step 2: Allocate the shared memory segment
    shmid = shmget(key, SHM_SIZE, 0666);
    if (shmid == -1) {
```

```
        perror("shmget failed");
        exit(1);
        }
        // Step 3: Attach the shared memory segment
        shm_ptr = (char *)shmat(shmid, NULL, 0);
        if (shm_ptr == (char *)(-1)) {
        perror("shmat failed");
        exit(1);
        }
        // Step 4: Print the shared memory contents
        printf("Received message: %s\n", shm_ptr);
        // Step 5: Detach shared memory segment
        if (shmdt(shm_ptr) == -1) {
        perror("shmdt failed");
        exit(1);
        }
        return 0;
}
```

Sample Output:



Result:

Hence, IPC using Shared Memory is executed successfully