Ex. No.: 7 **231901039**

Date: 19/2/2025

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 <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>
#include <unistd.h>

#define SHM_KEY 1234
#define SHM_SIZE 1024

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

```
    // Step 2: Allocate the shared memory
    shmid = shmget(SHM_KEY, SHM_SIZE,
IPC_CREAT | 0666);
    if (shmid < 0) {
        perror("shmget error");
        return 1;
    }

    // Step 3: Attach the shared memory
    shm_ptr = (char *)shmat(shmid, NULL, 0);
    if (shm_ptr == (char *)(-1)) {
        perror("shmat error");
        return 1;
    }

    // Step 4: Write to shared memory
    char message[] = "Hello from sender using
shared memory!";
    sprintf(shm_ptr, "%s", message);
    printf("Sender: Written to shared memory:
%s\n", message);

    // Step 5: Simulate delay
    sleep(5);

    // Step 6: Detach
    shmdt(shm_ptr);

    return 0;
}
```

receiver.c

```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <unistd.h>
#define SHM_KEY 1234
#define SHM_SIZE 1024

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

    // Step 2: Allocate the shared memory
    shmid = shmget(SHM_KEY, SHM_SIZE, 0666);
    if (shmid < 0) {
        perror("shmget error");
        return 1;
    }

    // Step 3: Attach the shared memory
    shm_ptr = (char *)shmat(shmid, NULL, 0);
    if (shm_ptr == (char *)(-1)) {
        perror("shmat error");
        return 1;
    }

    // Step 4: Read and display from shared
memory
    printf("Receiver: Received from shared
memory: %s\n", shm_ptr);

    // Step 5: Detach
    shmdt(shm_ptr);

    return 0;
}
```

Sample Output

Terminal 1

```
[root@localhost student]# gcc sender.c -o sender
[root@localhost student]# ./sender
```

Terminal 2

```
[root@localhost student]# gcc receiver.c -o receiver
[root@localhost student]# ./receiver
Message Received: Welcome to Shared Memory
[root@localhost student]#
```

Result:

Inter Process Communication (IPC) using shared memory between sender process and the receiver process has been implemented using C Program.