Ex. No.: 6

Date: 19.03.24

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.

Program Code:

sender.c

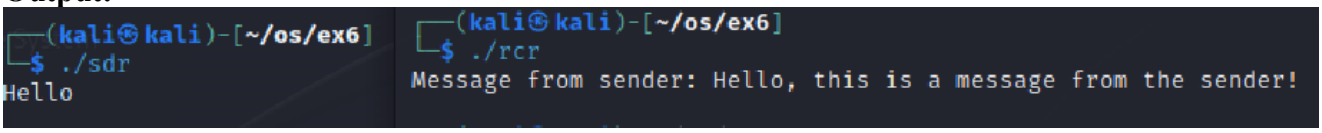
```
#include <stdio.h>
#include <stdlib.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>

#define SHM_KEY 12345
#define SHM_SIZE 1024

int main() {
   size_t shm_size = SHM_SIZE;

   int shm_id = shmget(SHM_KEY, shm_size, IPC_CREAT | S_IRUSR | S_IWUSR);
   if (shm_id == -1) {
      perror("shmget failed");
      exit(1);
   }

   char *shm_ptr = (char *)shmat(shm_id, NULL, 0);
   if (shm_ptr == (void *) -1) {
      perror("shmat failed");
      exit(1);
   }

   const char *message = "Hello, this is a message from the sender!";
   sprintf(shm_ptr, "%s", message);

   sleep(10);
   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/stat.h>
#include <unistd.h>
#include <string.h>

#define SHM_KEY 12345
#define SHM_SIZE 1024

int main() {
    size_t shm_size = SHM_SIZE;

    int shm_id = shmget(SHM_KEY, shm_size,
S_IRUSR | S_IWUSR);
    if (shm_id == -1) {
        perror("shmget failed");
        exit(1);
    }

    char *shm_ptr = (char *)shmat(shm_id, NULL,
0);
    if (shm_ptr == (void *) -1) {
        perror("shmat failed");
        exit(1);
    }

    printf("Message from sender: %s\n", shm_ptr);
    if (shmdt(shm_ptr) == -1) {
        perror("shmdt failed");
        exit(1);
    }
    if (shmctl(shm_id, IPC_RMID, NULL) == -1) {
        perror("shmctl failed");
        exit(1);
    }

    return 0;
}
```

Output:



Result:

The above program executed successfully and output got verified.