192311035

M.RAVI SAI VINAY

EXPERIMENT-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 in C, allowing two or more processes to communicate by sharing a common memory region.

ALGORITHM:-

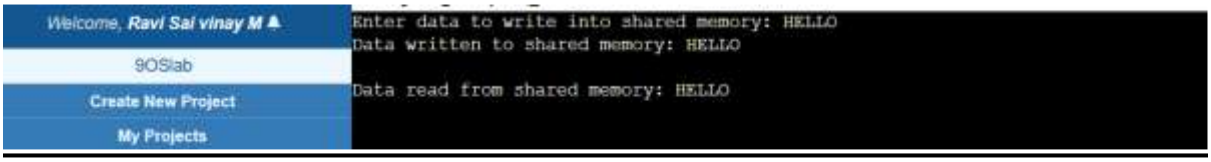
1. Create Shared Memory:
2. Use the shmget system call to allocate a shared memory segment.
3. Attach Shared Memory:
4. Use the shmat system call to attach the shared memory segment to the process's address space.
5. Read/Write to Shared Memory:
6. One process writes data into the shared memory.
7. Another process reads data from the shared memory.
8. Detach and Destroy Shared Memory:
9. Use shmdt to detach the shared memory from the process address space.
10. Use shmctl to remove the shared memory segment when it is no longer needed.

CODE:-

```
#include <stdio.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdlib.h>

#include <string.h>
```

```
#define SHM_SIZE 1024


int main() {

    key_t key = 1234;

    int shmid;

    char *shm;


    shmid = shmget(key, SHM_SIZE, 0666 | IPC_CREAT);

    if (shmid == -1) {

        perror("shmget failed");

        exit(1);

    }


    shm = (char *)shmat(shmid, NULL, 0);

    if (shm == (char *)(-1)) {

        perror("shmat failed");

        exit(1);

    }


    printf("Writer Process: Enter some data: ");

    fgets(shm, SHM_SIZE, stdin);

    printf("Writer Process: Data written to shared memory.\n");


    printf("Writer Process: Press Enter to detach and exit.");
    getchar();

    shmdt(shm);

    return 0;
}
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdlib.h>
#include <string.h>

#define SHM_SIZE 1024

int main() {
    key_t key = 1234;
    int shmid;
    char *shm;

    shmid = shmget(key, SHM_SIZE, 0666);
    if (shmid == -1) {
        perror("shmget failed");
        exit(1);
    }
    shm = (char *)shmat(shmid, NULL, 0);

    if (shm == (char *)(-1)) {

        perror("shmat failed");

        exit(1);

    }



    printf("Reader Process: Data read from shared memory: %s\n", shm);



    shmdt(shm);



    shmctl(shmid, IPC_RMID, NULL);



    return 0;

}
```

OUTPUT:-



```
Enter data to write into shared memory: HELLO
Data written to shared memory: HELLO

Data read from shared memory: HELLO
```

RESULT:-

- The program demonstrates the use of shared memory to communicate between two processes.
- The writer process writes data into shared memory, and the reader process reads data from it.
- After communication, the shared memory is detached and destroyed.