

Name: REVI THIMMA REDDY

Reg-No: 192325025

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

Aim

To demonstrate inter-process communication (IPC) using shared memory in C, where one process writes data to a shared memory segment, and another process reads it.

Algorithm

1. Create a shared memory segment using `shmget`.
2. Attach the shared memory segment to the address space of the process using `shmat`.
3. In one process:
 - Write data to the shared memory segment.
4. In another process:
 - Attach to the same shared memory.
 - Read the data from the shared memory segment.
5. Detach and delete the shared memory segment using `shmdt` and `shmctl`.

Procedure

1. Use `fork()` to create a parent and child process.
2. Parent writes data to the shared memory segment.
3. Child reads the data from the same shared memory segment.
4. Detach and clean up the shared memory after communication.

Code:

```

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

#define SHM_SIZE 1024

int main() {
    key_t key = ftok("shmfile", 65);
    int shmid = shmget(key, SHM_SIZE, 0666 | IPC_CREAT);
    if (shmid == -1) {
        perror("shmget failed");
        return 1;
    }

    char *shared_memory = (char *)shmat(shmid, NULL, 0);
    if (shared_memory == (char *)-1) {
        perror("shmat failed");
        return 1;
    }

    pid_t pid = fork();

    if (pid == 0) {
        sleep(1);
        printf("Child read: %s\n", shared_memory);
        shmdt(shared_memory);
    } else {
        strcpy(shared_memory, "Hello from parent!");
        wait(NULL);
        shmctl(shmid, IPC_RMID, NULL);
    }

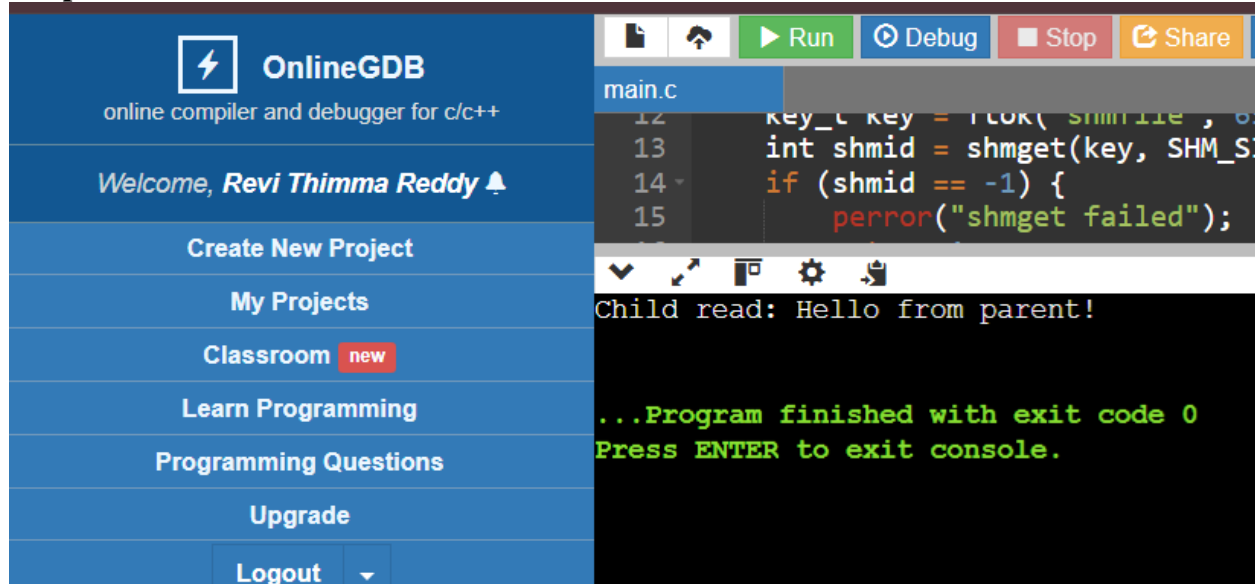
    return 0;
}

```

Result

- **Parent Process** writes the message "Hello from parent!" to the shared memory.
- **Child Process** reads the message from the shared memory and prints:
Child read: Hello from parent!

Output:



The screenshot displays the OnlineGDB web interface. On the left is a blue sidebar with the OnlineGDB logo, a welcome message for 'Revi Thimma Reddy', and navigation links: 'Create New Project', 'My Projects', 'Classroom' (with a 'new' badge), 'Learn Programming', 'Programming Questions', 'Upgrade', and a 'Logout' button. The main area on the right shows a C program in 'main.c' with lines 12-15 visible, involving shared memory operations. Below the code editor is a toolbar with icons for file operations, a 'Run' button, a 'Debug' button, a 'Stop' button, and a 'Share' button. The output console at the bottom shows the program's execution results.

```
12 key_t key = FLOCK( SHMFILE, 0);  
13 int shmid = shmget(key, SHM_SIZE, 0666);  
14 if (shmid == -1) {  
15     perror("shmget failed");  
16 }
```

Child read: Hello from parent!

...Program finished with exit code 0
Press ENTER to exit console.