192311035

M.RAVI SAI VINAY

EXPERIMENT-10

Illustrate the concept of inter-process communication using message queue with a C program.

AIM:-

To illustrate the concept of inter-process communication (IPC) using message queues in C. This allows two processes to communicate by sending and receiving messages through a message queue.

ALGORITHM:-

1. **Create a Message Queue:**
    - Use msgget to create a message queue identified by a key.
2. **Send Message (Writer Process):**
    - The writer process sends a message to the queue using msgsnd.
3. **Receive Message (Reader Process):**
    - The reader process retrieves the message from the queue using msgrcv.
4. **Message Queue Management:**
    - After reading the message, the reader process removes the message queue using msgctl to delete it.

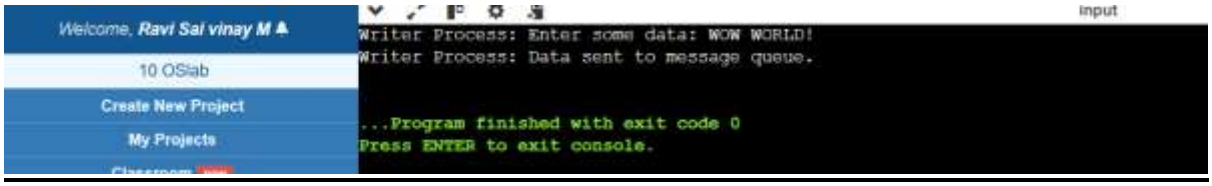
CODE:-

```
#include <stdio.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <stdlib.h>

#include <string.h>


#define MSG_SIZE 100
```

```
struct message {
    long msg_type;
    char msg_text[MSG_SIZE];
};

int main() {
    key_t key = 1234;
    int msgid;
    struct message msg;

    msgid = msgget(key, 0666 | IPC_CREAT);
    if (msgid == -1) {
        perror("msgget failed");
        exit(1);
    }

    printf("Writer Process: Enter some data: ");
    fgets(msg.msg_text, MSG_SIZE, stdin);
    msg.msg_type = 1;

    if (msgsnd(msgid, &msg, sizeof(msg), 0) == -1) {
        perror("msgsnd failed");
        exit(1);
```

```
    }

    printf("Writer Process: Data sent to message queue.\n");

    return 0;
}
```

OUTPUT:-



RESULT:-

- The program demonstrates IPC using message queues. The writer process sends a message to the queue, which the reader process retrieves and displays. After reading, the message queue is deleted, showing effective inter-process communication.