**Experiment:5-Illustrate the concept of inter-process communication using message queue with a C program.**

Aim:

The aim of this program is to illustrate Inter-Process Communication (IPC) using Message Queues in C. A message queue allows processes to communicate by sending and receiving messages in a queue. One process sends a message to the queue, and another process retrieves it from the queue. The messages are stored in the kernel and are accessed using a key.

Procedure:

1. Create a Message Queue:
   * Use the msgget() system call to create or access a message queue. The queue is identified by a unique key.
2. Send Message to the Queue:
   * A sender process uses the msgsnd() system call to send a message to the queue. The message is added to the queue.
3. Receive Message from the Queue:
   * A receiver process uses the msgrcv() system call to receive a message from the queue. The receiver waits for a message to be available in the queue.
4. Remove the Message Queue:
   * After the processes finish communication, the message queue is removed using the msgctl() system call to free the resources.

Steps in the Program:

1. Sender Process: Sends a message to the message queue.
2. Receiver Process: Receives a message from the message queue.

C Program Implementation:

Sender Program (writes message to message queue):

#include <stdio.h>

#include <stdlib.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <string.h>

#define MSG\_KEY 1234 // Message Queue key

struct msg\_buffer {

long msg\_type; // Message type (must be > 0)

char msg\_text[100]; // Message content

};

int main() {

int msgid;

struct msg\_buffer message;

// Create message queue

msgid = msgget(MSG\_KEY, 0666 | IPC\_CREAT);

if (msgid == -1) {

perror("msgget failed");

exit(1);

}

// Prepare message to send

message.msg\_type = 1; // Message type is 1 (can be any positive number)

printf("Enter the message to send: ");

fgets(message.msg\_text, sizeof(message.msg\_text), stdin);

// Send the message to the queue

if (msgsnd(msgid, &message, sizeof(message), 0) == -1) {

perror("msgsnd failed");

exit(1);

}

printf("Message sent: %s", message.msg\_text);

return 0;

}

Output:

A screenshot of a computer screen

Description automatically generated