C Programming Basics
Lab Session: IPC using Message Queue with fork()
Course: Operating Systems
Date: 31.05.2025

Program: IPC using Message Queue with fork()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ipc.h>      // For IPC key generation
#include <sys/msg.h>      // For message queue functions
#include <unistd.h>       // For fork()

#define MAX 10            // Max size of message text

// Structure for message queue
struct mesg_buffer {
    long mesg_type;
    char mesg_text[100];
} message;

int main() {
    int f = fork();  // Create child process

    if (f == 0) {
        // Child - Receive
        key_t key;
```

```
    int msgid;

    key = ftok("nuzha1", 130);
    msgid = msgget(key, 0666 | IPC_CREAT);
    msgrcv(msgid, &message, sizeof(message.mesg_text), 1, 0);

    // Parse received message
    char name[30], regno[30], age[10];
    sscanf(message.mesg_text, "%s %s %s", name, regno, age);

    printf("\nReceived Name : %s\n", name);
    printf("Received RegNo : %s\n", regno);
    printf("Received Age : %s\n", age);

    msgctl(msgid, IPC_RMID, NULL);  // Remove message queue
} else {
    // Parent - Send
    key_t key;
    int msgid;
    char name[30], regno[30], age[10];

    key = ftok("nuzha1", 130);
    msgid = msgget(key, 0666 | IPC_CREAT);

    printf("Enter Name: ");
    scanf("%s", name);
```

```
        printf("Enter RegNo: ");
        scanf("%s", regno);
        printf("Enter Age: ");
        scanf("%s", age);

        message.mesg_type = 1;
        snprintf(message.mesg_text, sizeof(message.mesg_text), "%s %s %s", name, regno, age);

        msgsnd(msgid, &message, sizeof(message.mesg_text), 0);
        printf("\nMessage sent successfully.\n");
    }

    return 0;
}
```

Fedora Output:

```
[2021ict108@fedora ~]$ vi nuzha1.c
[2021ict108@fedora ~]$ gcc nuzha1.c -o nuzha1
[2021ict108@fedora ~]$ ./nuzha1
Enter Name: nuzha
Enter RegNo: 2021ict108
Enter Age: 24

Message sent successfully.

Received Name : nuzha
```

Received RegNo : 2021ict108

Received Age : 24

[2021ict108@fedora ~]$

Explanation:

- This program demonstrates **IPC using System V message queues** and **process creation with fork()**.
- The **parent process** reads user input (Name, RegNo, Age) and sends it through the message queue.
- The **child process** reads from the queue, parses the message, and prints the received data.
- ftok() generates a unique key.
- msgget() creates/accesses the queue.
- msgsnd() and msgrcv() handle communication.
- msgctl() removes the message queue after use.