Sender Process (sender.c)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#define MAX\_TEXT 512

struct message {

long msg\_type;

char msg\_text[MAX\_TEXT];

};

int main() {

key\_t key = ftok("msgqueuefile", 65); // Generate unique key

int msgid = msgget(key, 0666 | IPC\_CREAT); // Create message queue and return id

struct message msg;

msg.msg\_type = 1;

printf("Write Data: ");

fgets(msg.msg\_text, MAX\_TEXT, stdin); // Get user input

msgsnd(msgid, &msg, sizeof(msg), 0); // Send message

printf("Data sent: %s\n", msg.msg\_text);

return 0;

}

C:\Users\venka\OneDrive\Pictures\Screenshots\Screenshot 2024-12-06 124052.png

Receiver Process (receiver.c)

#include <stdio.h>

#include <stdlib.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#define MAX\_TEXT 512

struct message {

long msg\_type;

char msg\_text[MAX\_TEXT];

};

int main() {

key\_t key = ftok("msgqueuefile", 65); // Generate unique key

int msgid = msgget(key, 0666 | IPC\_CREAT); // Get message queue id

struct message msg;

msgrcv(msgid, &msg, sizeof(msg), 1, 0); // Receive message

printf("Data received: %s\n", msg.msg\_text);

msgctl(msgid, IPC\_RMID, NULL); // Destroy the message queue

return 0;

}

