#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <unistd.h>

#define MAX 100

// Message structure

struct msg\_buffer {

long msg\_type; // must be of type long

char msg\_text[MAX];

};

int main() {

struct msg\_buffer message;

key\_t key;

int msgid;

// Generate unique key

key = ftok("progfile", 65);

// Create message queue and return identifier

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

if (fork() == 0) {

// Child process -> Sender

message.msg\_type = 1;

printf("Sender: Enter a message: ");

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

// Send message

msgsnd(msgid, &message, sizeof(message.msg\_text), 0);

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

} else {

// Parent process -> Receiver

sleep(2); // Ensure sender runs first

// Receive message

msgrcv(msgid, &message, sizeof(message.msg\_text), 1, 0);

printf("Receiver: Message received -> %s", message.msg\_text);

// Destroy the message queue

msgctl(msgid, IPC\_RMID, NULL);

}

return 0;

}

