

Самарский национальный исследовательский университет  
имени академика С.П. Королева  
Естественнонаучный институт  
Механико-математический факультет

Кафедра информатики и вычислительной математики

Отчет по лабораторной работе №8  
Итоговая  
Индивидуальное задание

Выполнил:  
студент группы 4445-020303D  
Л.М. Тарасов

Проверил:  
Д.А.Левкин

Самара-2023

## **Тема: Разработка мультипроцессных приложений.**

**Цель работы:** Закрепление работы с функциями ОС РВ для запуска параллельных процессов и организации межпроцессного взаимодействия посредством сообщений.

### **Задание**

Разработать приложение, состоящее из трех взаимодействующих процессов. Требуется написать три программных модуля: M1, M2, M3. На базе модуля M1 из shell запускается стартовый процесс P1(M1).

Процесс P1 создает канал и, используя функцию семейства `spawn*`(), запускает процесс P2(M2), передавая ему в качестве параметра `chid` созданного канала, и переходит в состояние приема сообщений от P2 по своему каналу.

Процесс P2 создает свой канал и, используя функцию семейства `spawn*`(), запускает процесс P3(M3), передавая ему в качестве параметра `chid` созданного канала. Устанавливает соединение с каналом процесса P1, отправляет ему сообщение о `pid` процесса P3 и переходит в состояние приема сообщений по созданному каналу от P3.

Процесс P3 создает свой канал, устанавливает соединение с каналом процесса P2 и посылает ему сообщение "P3 загружен". Получив ответ, посылает ему `chid` своего канала и переходит в состояние приема сообщений по своему каналу.

Процесс P2, приняв первое сообщение, отправляет его процессу P1, получив ответ от P1, принимает `chid` от процесса P3, посылает ему ответ и передает `chid` процессу P1. Далее выдает на терминал "P2 загружен" и переходит в состояние приема сообщений по своему каналу от P3.

Процесс P1, получив первое сообщение от P2, выдает его на экран терминала, посылает ответ процессу P2 и принимает второе сообщение, устанавливает соединение с каналом процесса P3 и посылает по нему сообщение "stop", после ответа переходит в ожидание сообщения по своему каналу.

Процесс P3, получив "stop", отправляет его процессу P2, печатает "stop P3" и завершается.

Процесс P2, получив "stop", отправляет его процессу P1, печатает "stop P2" и завершается.

Процесс P1, получив "stop", печатает "stop P1" и завершается.

## Листинг модуля M1

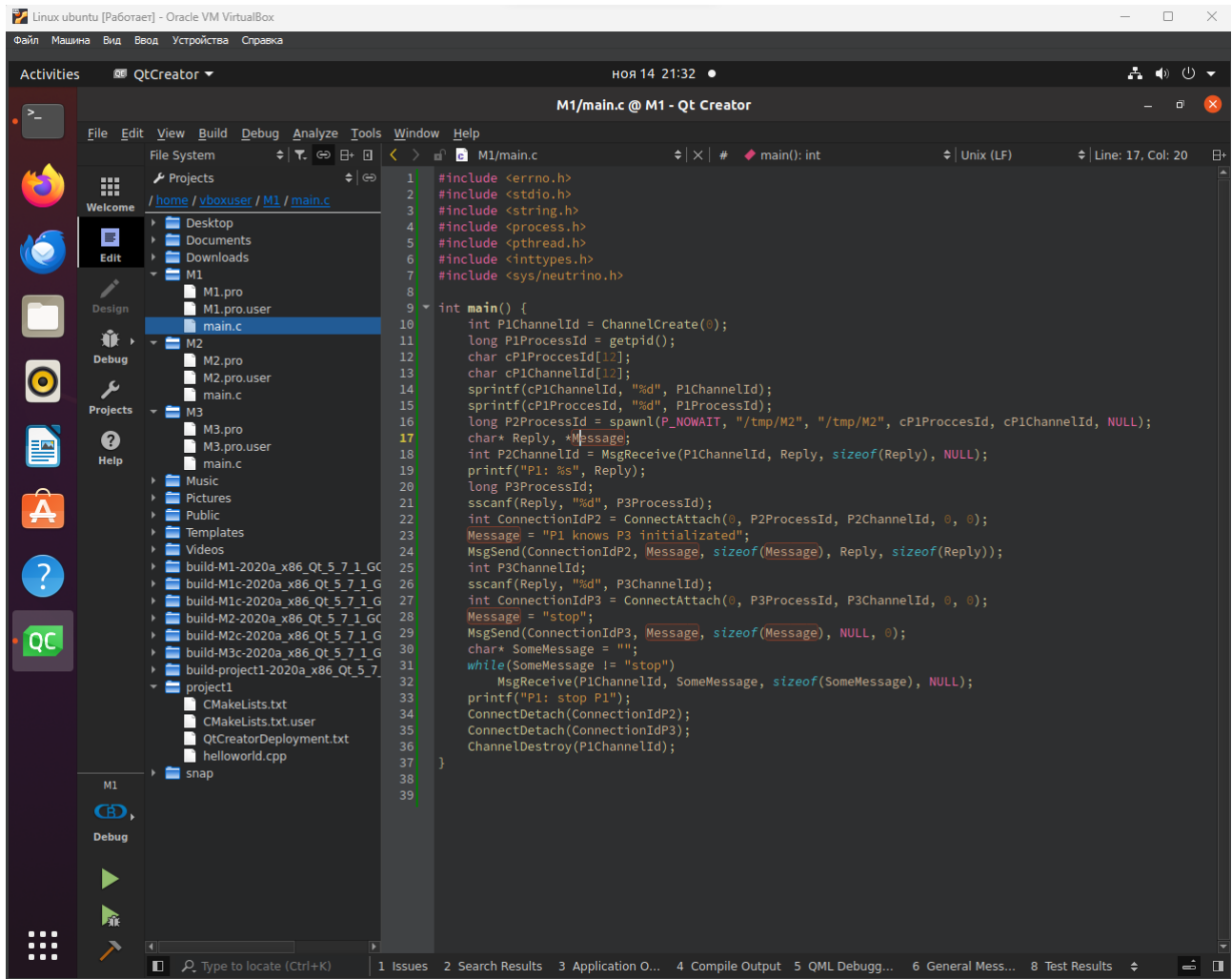


Рисунок 1 – M1 в QtCreator

```
#include <errno.h>
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <process.h>
```

```
#include <pthread.h>
```

```
#include <inttypes.h>
```

```
#include <sys/neutrino.h>
```

```
int main() {
```

```
    int P1ChannelId = ChannelCreate(0);
```

```
    long P1ProcessId = getpid();
```

```

char cP1ProccesId[12];

char cP1ChannelId[12];

sprintf(cP1ChannelId, "%d", P1ChannelId);

sprintf(cP1ProccesId, "%d", P1ProcessId);

long P2ProcessId = spawnl(P_NOWAIT, "/tmp/M2", "/tmp/M2", cP1ProccesId, cP1ChannelId,
NULL);

char* Reply, *Message;

int P2ChannelId = MsgReceive(P1ChannelId, Reply, sizeof(Reply), NULL);

printf("P1: %s", Reply);

long P3ProcessId;

sscanf(Reply, "%d", P3ProcessId);

int ConnectionIdP2 = ConnectAttach(0, P2ProcessId, P2ChannelId, 0, 0);

Message = "P1 knows P3 initializated";

MsgSend(ConnectionIdP2, Message, sizeof(Message), Reply, sizeof(Reply));

int P3ChannelId;

sscanf(Reply, "%d", P3ChannelId);

int ConnectionIdP3 = ConnectAttach(0, P3ProcessId, P3ChannelId, 0, 0);

Message = "stop";

MsgSend(ConnectionIdP3, Message, sizeof(Message), NULL, 0);

char* SomeMessage = "";

while(SomeMessage != "stop")

    MsgReceive(P1ChannelId, SomeMessage, sizeof(SomeMessage), NULL);

printf("P1: stop P1");

ConnectDetach(ConnectionIdP2);

ConnectDetach(ConnectionIdP3);

ChannelDestroy(P1ChannelId);

}

```

## Листинг модуля M2

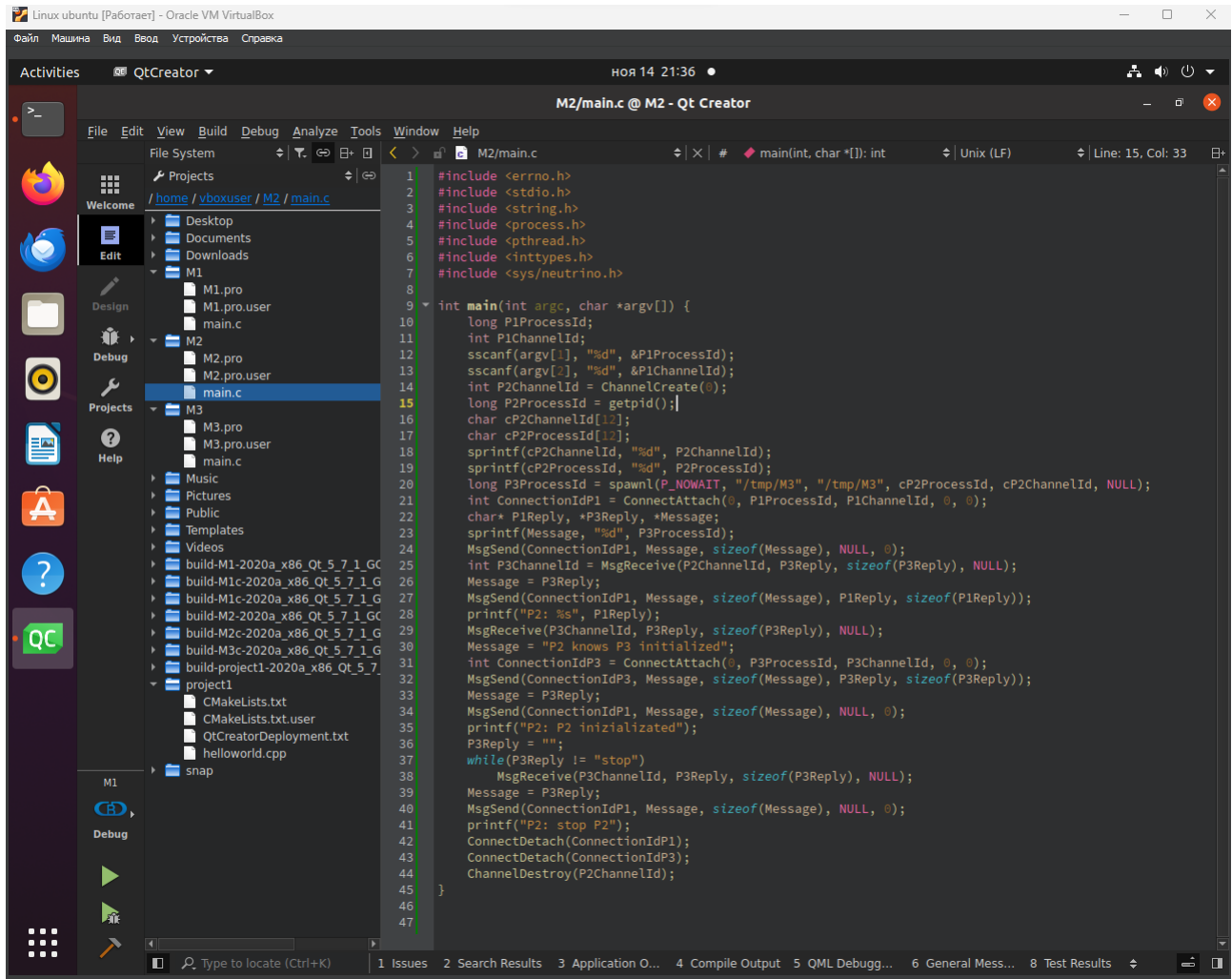


Рисунок 2 – M2 в QtCreator

```
#include <errno.h>
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <process.h>
```

```
#include <pthread.h>
```

```
#include <inttypes.h>
```

```
#include <sys/neutrino.h>
```

```
int main(int argc, char *argv[]) {
```

```
    long P1ProcessId;
```

```
    int P1ChannelId;
```

```

sscanf(argv[1], "%d", &P1ProcessId);

sscanf(argv[2], "%d", &P1ChannelId);

int P2ChannelId = ChannelCreate(0);

long P2ProcessId = getpid();

char cP2ChannelId[12];

char cP2ProcessId[12];

sprintf(cP2ChannelId, "%d", P2ChannelId);

sprintf(cP2ProcessId, "%d", P2ProcessId);

long P3ProcessId = spawnl(P_NOWAIT, "/tmp/M3", "/tmp/M3", cP2ProcessId, cP2ChannelId,
NULL);

int ConnectionIdP1 = ConnectAttach(0, P1ProcessId, P1ChannelId, 0, 0);

char* P1Reply, *P3Reply, *Message;

sprintf(Message, "%d", P3ProcessId);

MsgSend(ConnectionIdP1, Message, sizeof(Message), NULL, 0);

int P3ChannelId = MsgReceive(P2ChannelId, P3Reply, sizeof(P3Reply), NULL);

Message = P3Reply;

MsgSend(ConnectionIdP1, Message, sizeof(Message), P1Reply, sizeof(P1Reply));

printf("P2: %s", P1Reply);

MsgReceive(P3ChannelId, P3Reply, sizeof(P3Reply), NULL);

Message = "P2 knows P3 initialized";

int ConnectionIdP3 = ConnectAttach(0, P3ProcessId, P3ChannelId, 0, 0);

MsgSend(ConnectionIdP3, Message, sizeof(Message), P3Reply, sizeof(P3Reply));

Message = P3Reply;

MsgSend(ConnectionIdP1, Message, sizeof(Message), NULL, 0);

printf("P2: P2 inzialized");

P3Reply = "";

while(P3Reply != "stop")

    MsgReceive(P3ChannelId, P3Reply, sizeof(P3Reply), NULL);

Message = P3Reply;

```

```
MsgSend(ConnectionIdP1, Message, sizeof(Message), NULL, 0);  
printf("P2: stop P2");  
ConnectDetach(ConnectionIdP1);  
ConnectDetach(ConnectionIdP3);  
ChannelDestroy(P2ChannelId);  
}
```

## Листинг модуля M3

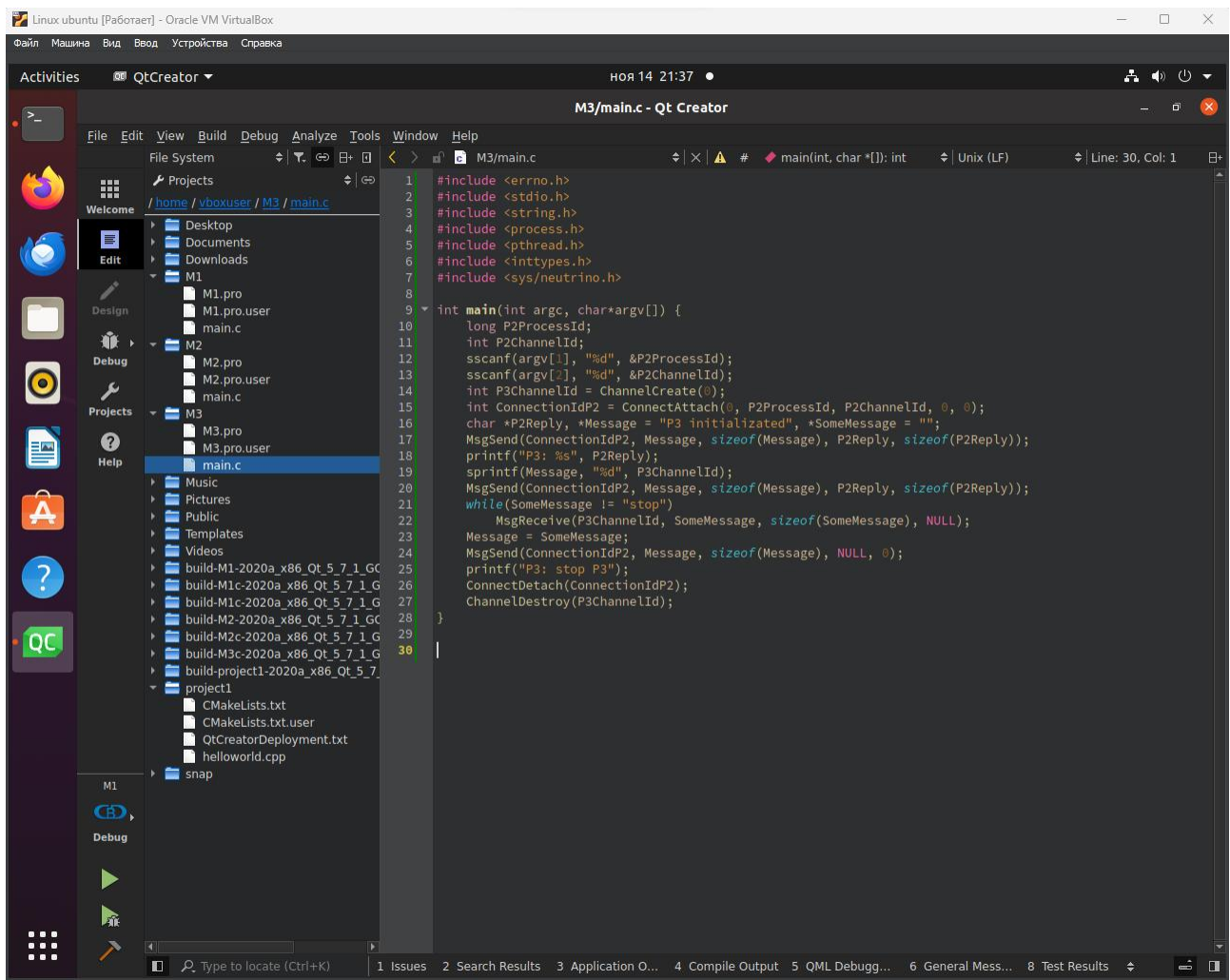


Рисунок 3 – M3 в QtCreator

```
#include <errno.h>
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <process.h>
```

```
#include <pthread.h>
```

```
#include <inttypes.h>
```

```
#include <sys/neutrino.h>
```

```
int main(int argc, char*argv[]) {
```

```
    long P2ProcessId;
```

```
    int P2ChannelId;
```



```
sscanf(argv[1], "%d", &P2ProcessId);

sscanf(argv[2], "%d", &P2ChannelId);

int P3ChannelId = ChannelCreate(0);

int ConnectionIdP2 = ConnectAttach(0, P2ProcessId, P2ChannelId, 0, 0);

char *P2Reply, *Message = "P3 initialized", *SomeMessage = "";

MsgSend(ConnectionIdP2, Message, sizeof(Message), P2Reply, sizeof(P2Reply));

printf("P3: %s", P2Reply);

sprintf(Message, "%d", P3ChannelId);

MsgSend(ConnectionIdP2, Message, sizeof(Message), P2Reply, sizeof(P2Reply));

while(SomeMessage != "stop")

    MsgReceive(P3ChannelId, SomeMessage, sizeof(SomeMessage), NULL);

Message = SomeMessage;

MsgSend(ConnectionIdP2, Message, sizeof(Message), NULL, 0);

printf("P3: stop P3");

ConnectDetach(ConnectionIdP2);

ChannelDestroy(P3ChannelId);

}
```

## Результат выполнения программы

```
P1: 5238857
P2: P1 knows P3 initialized
P3: P2 knows P3 initialized
P2: P2 initialized
P3: stop P3
P2: stop P2
P1: stop P1
# -
```