Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Лабораторная работа №3

«Электростанция»

Выполнили: студентки 4 курса

ИВТ, гр. ИП-813

Захарова К.Ю.

Колосова Я.В.

Проверил: преподаватель кафедры ПМиК

Белевцова Е.A.

Новосибирск, 2021 г.

Оглавление

[Задание на лабораторную работу 3](#_Toc85234665)

[Выполнение 4](#_Toc85234666)

[Вывод 13](#_Toc85234667)

# **Задание на лабораторную работу**

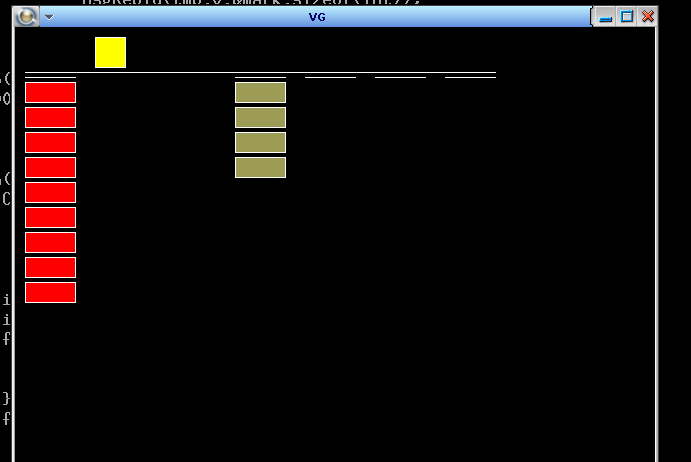
1. Электростанция состоит из следующих элементов: хранилище топлива (1 шт.), транспортное средство (1 шт.), котлы (4 шт.). Элементы станции работают параллельно, каждый по своей программе (что может быть реализовано с помощью нитей). Транспортное средство доставляет топливо из хранилища к котлам. Топливо имеет различные марки (от 1 до 10). Топливо марки 10 горит в котле 10 с (условно), в то время как топливо марки 1 горит всего 1 с. Необходимо написать программу, моделирующую работу электростанции и показывающую на экране процесс ее функционирования.
2. А теперь добавьте второе транспортное средство.
3. (использование импульсов) Регулируя скорости работы элементов электростанции, вы можете создать ситуацию, когда котлы будут простаивать из-за низкой скорости подвоза топлива. Создайте такую ситуацию. Теперь сделайте так, чтобы топливо подвозилось к котлам заранее, до момента их полной остановки. Это можно реализовать, если котлы будут сообщать о том, что топливо скоро кончится (например, его осталось на 2 с работы). Ясно, что котлы могут это сделать с помощью импульса, т.к. обычное сообщение их заблокировало бы, в то время как они должны продолжать работать
4. А теперь сделайте сетевой вариант разработанной программы. Пусть теперь хранилище работает на одной машине, котлы на другой, а транспортные средства переносят топливо между этими двумя машинами. Вам понадобится разделить вашу программу на две, которые вы будете запускать на разных узлах сети. Чтобы не надоедать соседу с просьбами о запуске вашей программки, которая к тому же будет постоянно подвешивать его компьютер, отладку можно вести на своем компьютере. Напомним, что имя узла содержится в переменной HOSTNAME.

# **Выполнение**

**1)** При выполнении задания были задействованы нити (потоки) из библиотеки <pthread> и функции обмена сообщениями между ними. Так как сообщения нельзя передать напрямую были задействованы специальные каналы и соединения. Для создания такого канала используется функция ChannelCreate(unsigned flags), принимающая лишь флаг, потом уже можно вызвать функции MsgRecieve((int chid, void \* msg, int bytes, struct \_msg\_info \* info) – принимает chid – id канала который мы получаем после вызова ChannelCreate(), msg – указатель на буфер сообщения, bytes – размер данного буфера, info – указатель на структуру, в которой может храниться доп. информация о сообщении.

Чтобы принимающий поток мог получить сообщение необходимо присоединиться к каналу отдающему через ConnectAttach(uint32\_t nd, \_t pid, int chid, index, int flags), а потом уже вызывать MsgSend(int coid, const void\* smsg, int sbytes, \* rmsg, int rbytes) с параметрами: coid – ID, полученный от ConnectAttach. smsg – указатель на буфер сообщения, sbytes – кол-во байтов на отправку, rmsg – указатель на буфер с ответом, rbytes – размер буфера ответа. В коде работа транспорта, склада и котлов реализовано в разных нитях (потоках), графическое представление работы программы реализовано через библиотеку VinGraph.h.

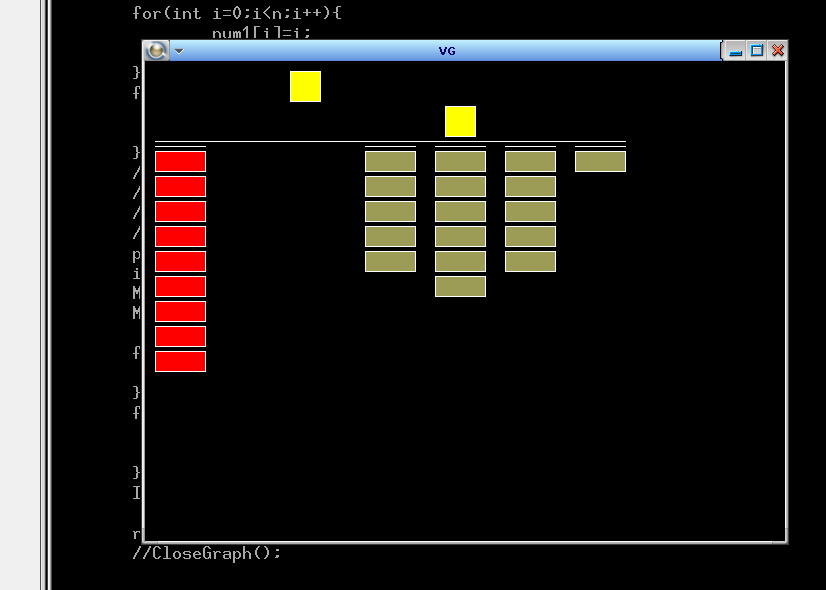
**Выполнение программы:**



**2)** А теперь добавьте второе транспортное средство.

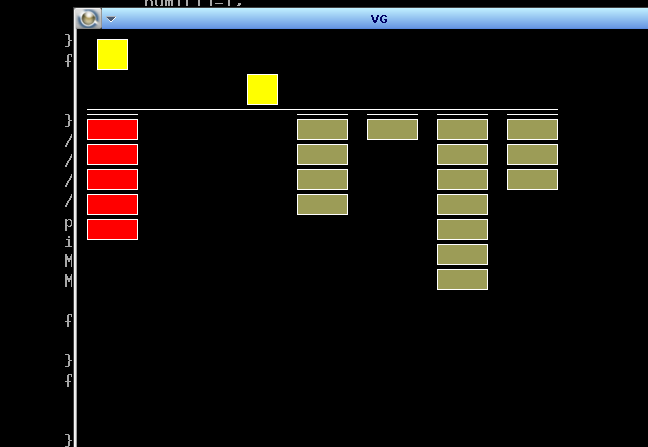
Второе транспортное средство легко добавить через добавление дополнительного потока с автомобилем в функции main() c помощью повторения вызова функции - pthread\_create(0, 0, chan1, 0);

**Выполнение программы:**



**3)** Для реализации данного задания были использованы pulse-методы, которые оповещают потоки о необходимости доставки для котлов в которых топлива осталось меньше чем на 3 секунды, чтобы отправить пульс используется функция MsgSendPulse(int coid, int priority, int code, int value) – отправляет импульс к потоку.

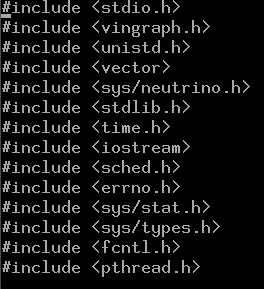
**Выполнение программы:**

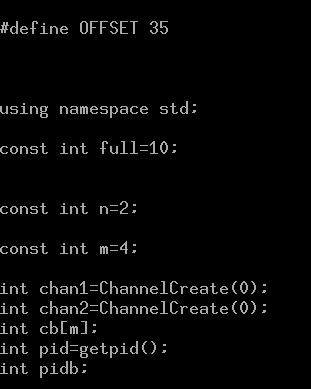


Как видно,транспортное средство уже поехалок котлу, в котором топливу осталось гореть меньше 3-х секунд.

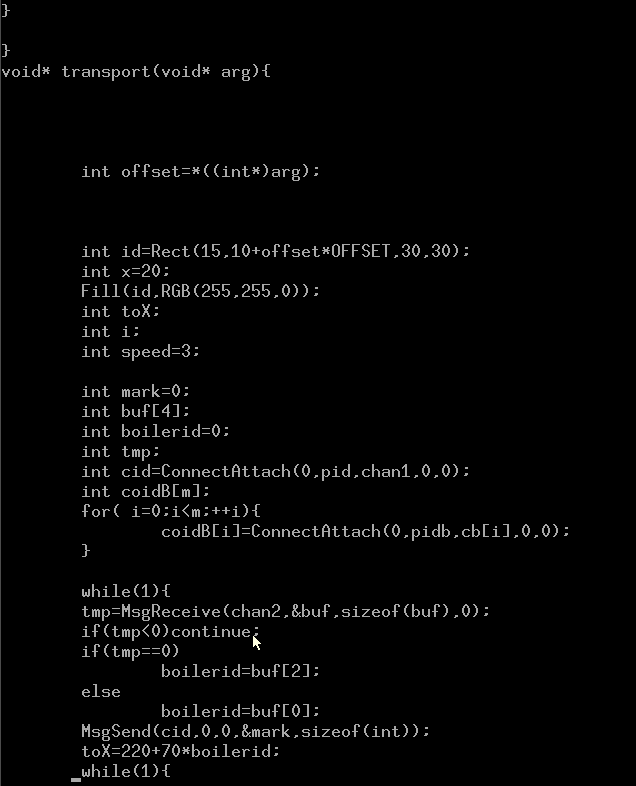
**Код программы:**

Запуск: cc lab3\_1.cpp –l vg –l m (далее ./a.out)



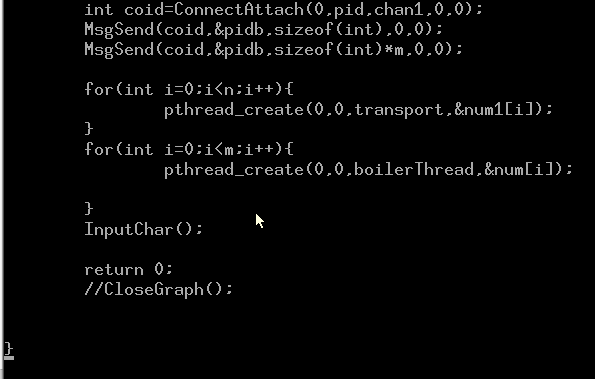








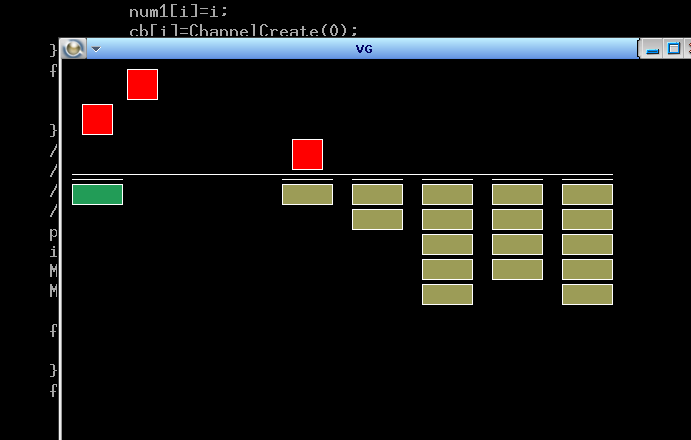




**4)** Для реализации этого задания предыдущая лабораторная была разбита на 2 файла – в одном были реализованные функции для транспорта и станции, а в другом функция для котлов. Передача многокомпонентных сообщений осуществляется через MsgSendv-посылает многокомпонентное сообщение и ждёт многокомпонентный ответ. Возвращает значение параметра stat из соответствующей функции передачи ответа (MsgReceivev,MsgReplyv). Для реализации сети вызывались два процесса: на узле “qnx425-1” и на узле “qnx425-2”.

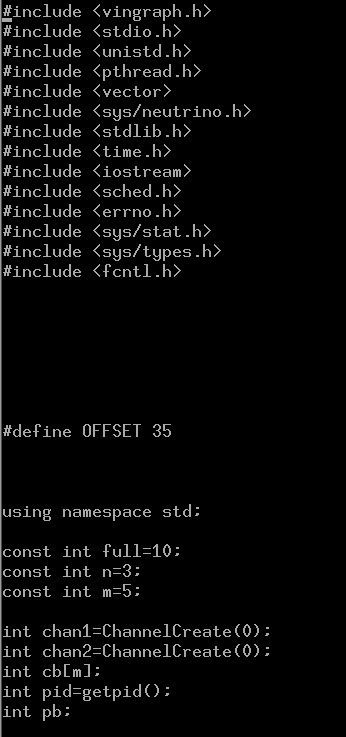
Запуск программы: в одном окне терминала (ttyp0) запускается файл, где хранится функция транспорта и станции. (сс lab3\_2.cpp –l vg –l m (далее ./a.out)). В другом окне терминала(ttyp1) запускается файл с функций котлов (аналогично первому).

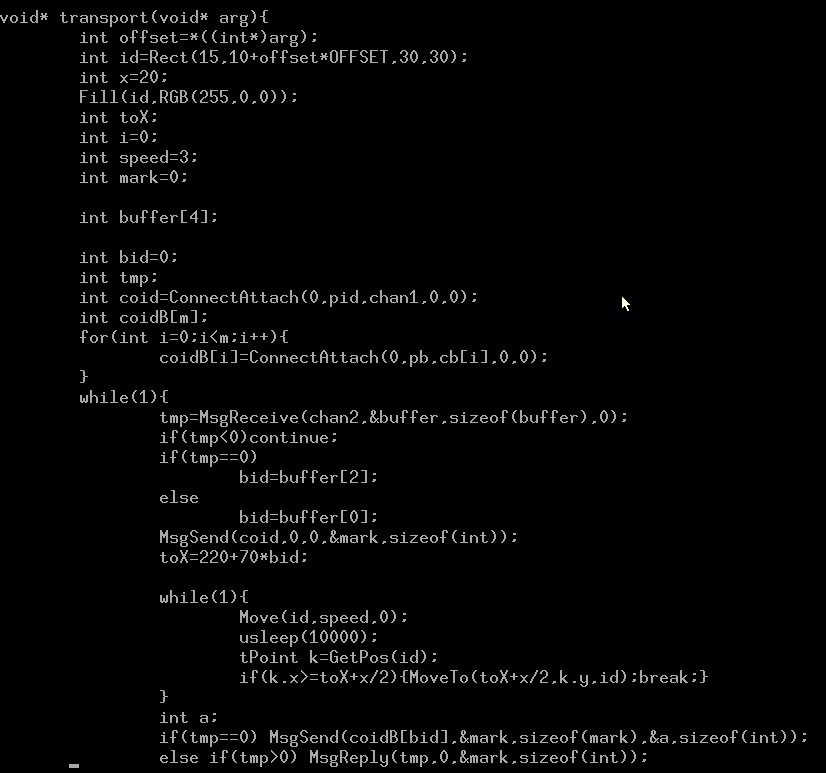
**Выполнение программы:**

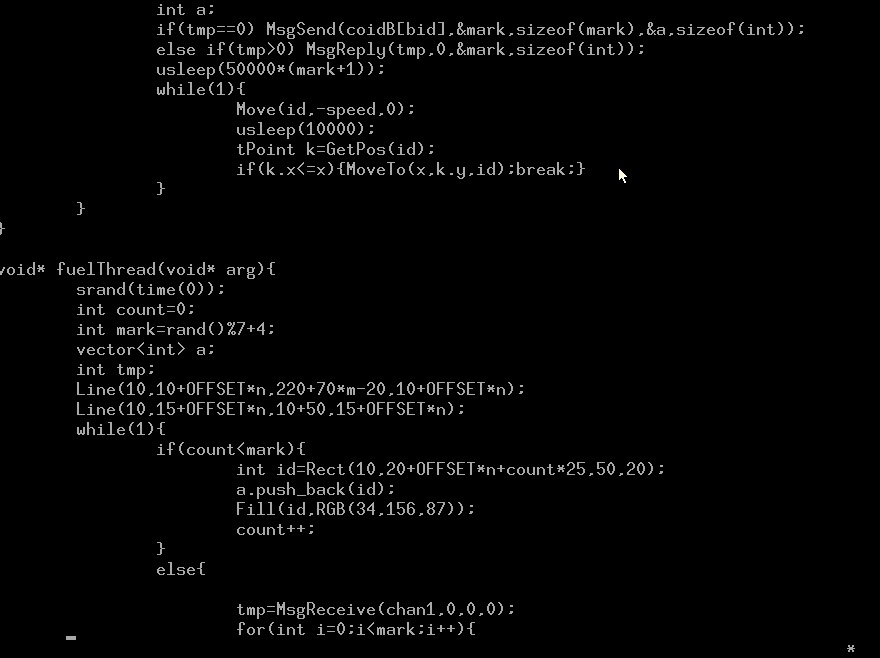


**Код программы:**

Lab3\_2.cpp

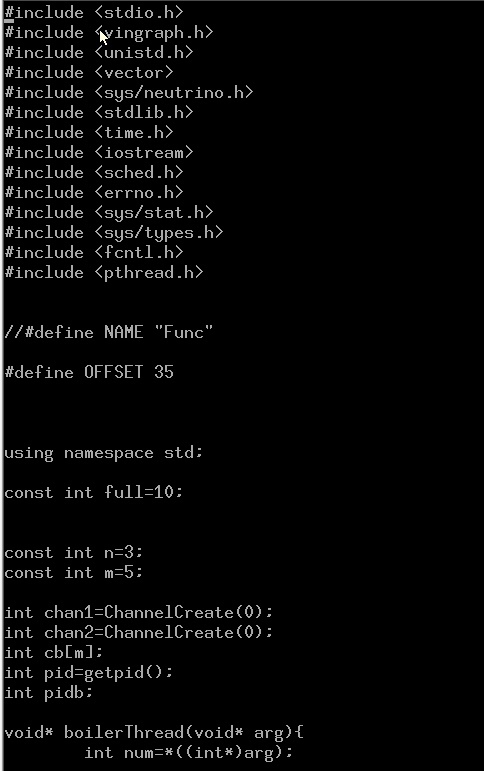




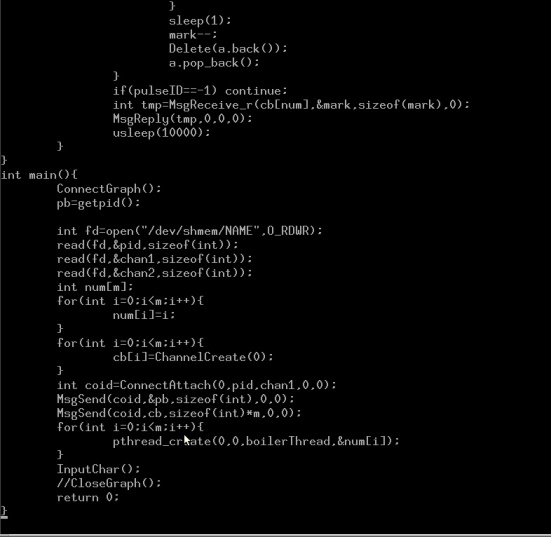




Lab3\_3.cpp







# **Вывод**

В этой лабораторной работе мы узнали о взаимодействии между нитями с помощью сообщений, применили знания об импульсах и сетях на практике.