Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Савостин Олег

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Цель данной лабораторной работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 2 Задание

1. Изучение программы Hello world!  
2. Транслятор NASM  
3. Расширенный синтаксис командной строки NASM  
4. Компоновщик LD  
5. Запуск исполняемого файла  
6. Выполнение заданий самостоятельной работы

# 3 Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины(ЭВМ) являются центральный процессор, память и периферийные устройства.Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской (системной) плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора (ЦП) входят следующие устройства: • арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; • устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; • регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; Важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ). ОЗУ — это быстродействующее энергозависимое запоминающее устройство, которое на- прямую взаимодействует с узлами процессора, предназначенное для хранения программ и данных, с которыми процессор непосредственно работает в текущий момент. Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня, таких как C/C++, Perl, Python и пр.

# 4 Выполнение лабораторной работы

## 4.1 Изучение программы Hello world!

Для начала выполнения лабораторной работы, я перехожу в каталог “lab04” с помощью утилиты “cd”(рис. 1).

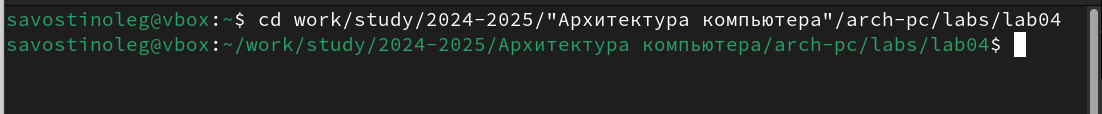


Рис. 1: Переход в нужную папку с помощью cd

Затем я создаю новый файл под названием “hello.asm”(рис. 2) и открываю новый файл с помощью gedit(рис. 3)

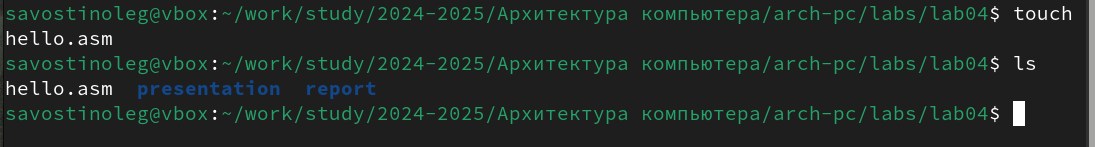


Рис. 2: Создание нового файла “hello.asm”

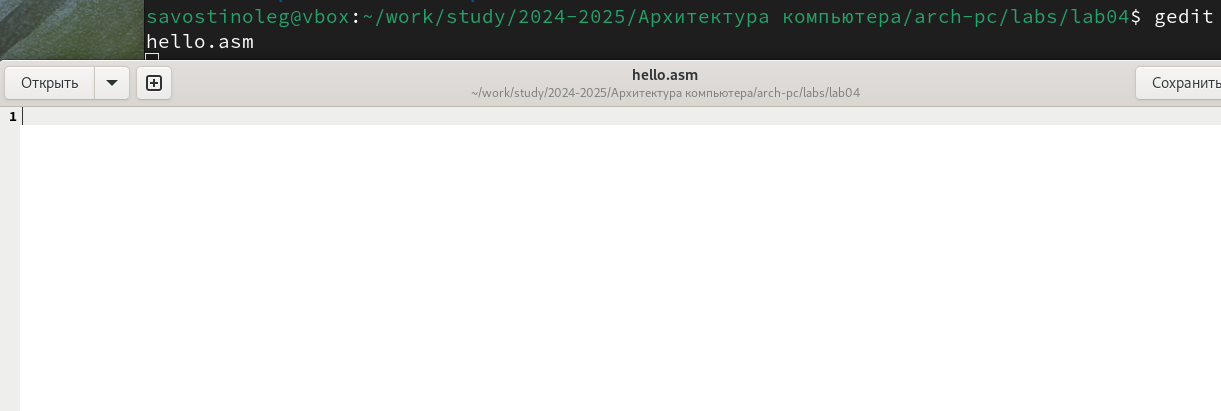


Рис. 3: Открытый ранее созданный мною файл.

Теперь, я ввожу текст в данный файл, который был предоставлен в файле лабораторной работы (4)

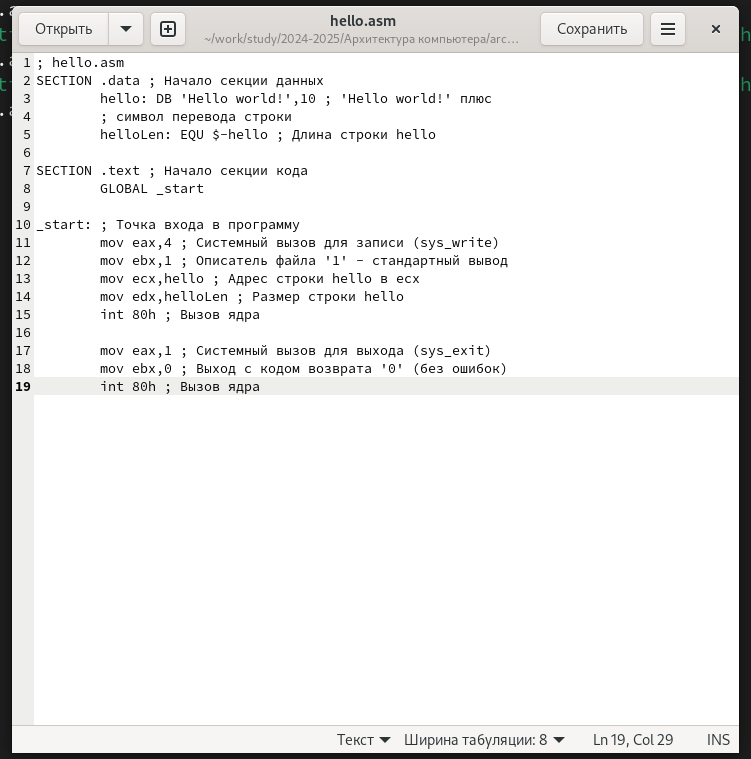


Рис. 4: Заполненный файл hello.asm

## 4.2 Транслятор NASM

NASM превращает текст программы в объектный код. Для компиляции текста в файле “hello.asm” я ввожу команду “nasm -f elf hello.asm” (5)

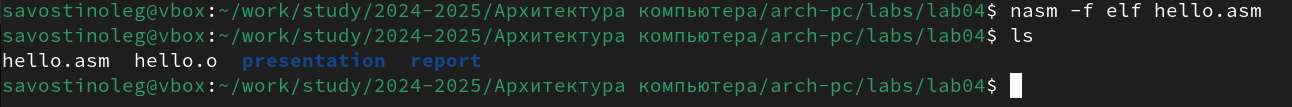


Рис. 5: Компиляция файла и проверка на успех совершения команды.

Компиляция прошла успешно и как видно на изображении, появился файл hello.o

## 4.3 Расширенный синтаксис командной строки NASM

Теперь я скомпилирую исходный файл hello.asm в obj.o и кроме того, будет создан файл листинга list.lst. Затем проверяю на правильность выполнения команды.(6)

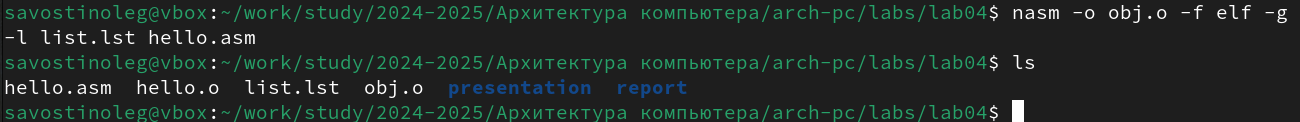


Рис. 6: Компиляция файла в obj.o, формирование файла list.lst и проверка на успех совершения команды.

## 4.4 Компоновщик LD

Теперь, я передаю объектный файл на обработку компоновщику с помощью команды ld -m elf\_i386 hello.o -o hello и проверяю если исполняемый файл hello был создан.(7)

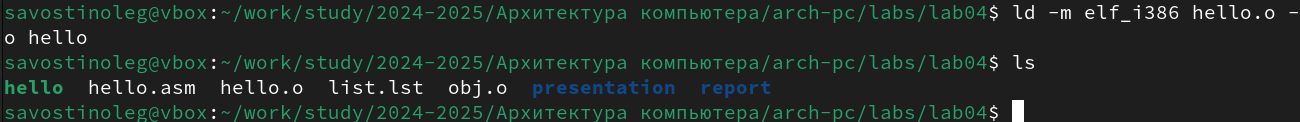


Рис. 7: Обработка файла и проверка на наличие исполняемого файла.

Ключ -o с последующим значением задаёт в данном случае имя создаваемого исполняемого файла и чтобы проверить это я ввожу команду: ld -m elf\_i386 obj.o -o main , после которой появляется файл main.(8)

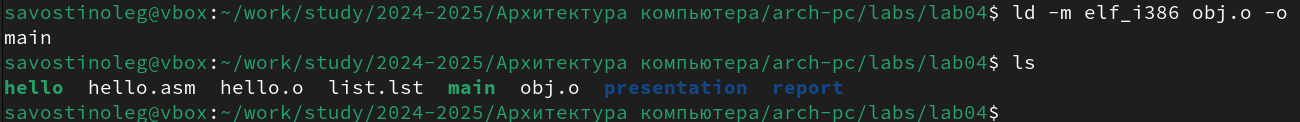


Рис. 8: Выполнение команды и проверка на наличие файла main.

## 4.5 Запуск исполняемого файла

И теперь я запускаю на выполнение файл, вводя команду в командной строке: ./hello , что выводит на экран Hello world!(9)

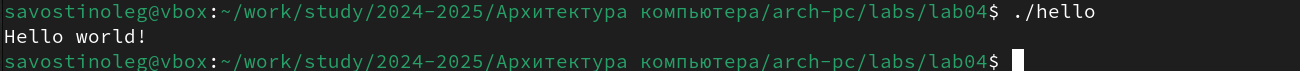


Рис. 9: Выполнение созданного исполняемого файла

## 4.6 Выполнение заданий самостоятельной работы

1. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создаю копию файла hello.asm с именем lab4.asm и проверяю на его наличие(10)

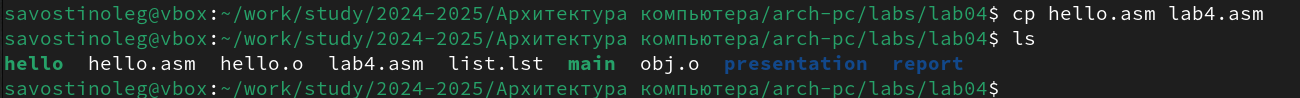


Рис. 10: Копирование файла с именем lab4.asm

1. С помощью программного текстового редактора вношу изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с моими фамилией и именем(11)



Рис. 11: Изменение текста в lab4.asm

1. Транслирую полученный текст программы lab4.asm в объектный файл. Затем выполняю компоновку объектного файла и запускаю получившийся исполняемый файл(12)

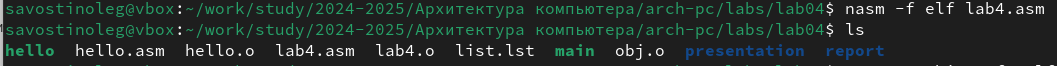


Рис. 12: Трансляция из asm. в .o

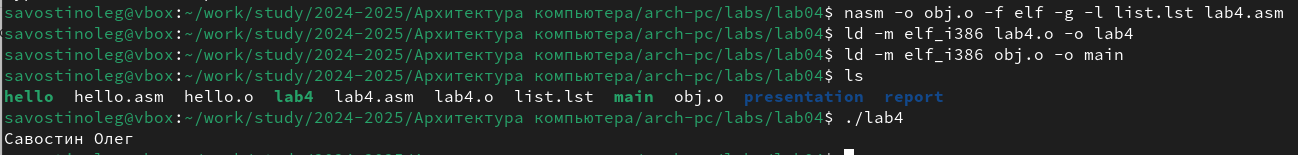


Рис. 13: Вывод моих фамилии и имени в терминале.

1. Так как файлы hello.asm и lab4.asm уже находятся в нужном репозитории, я их только отправляю на Github.(14)

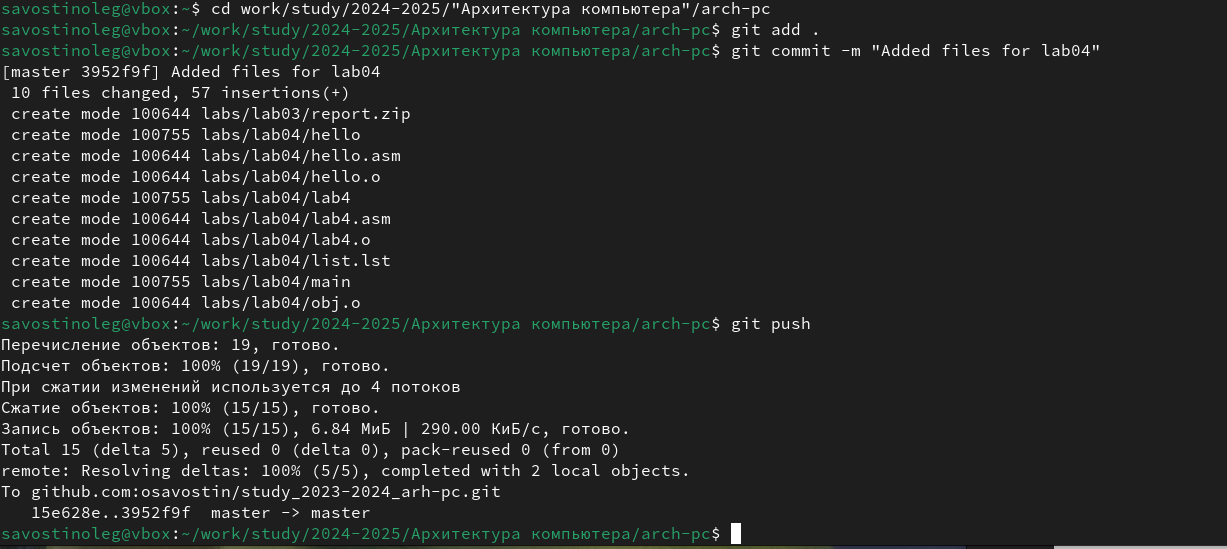


Рис. 14: Отправка файлов на Github

# 5 Выводы

В ходе данной лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# Список литературы

1. [Архитектура ЭВМ - РУДН](https://esystem.rudn.ru/pluginfile.php/2089533/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%964.%20%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B8%20%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%20%D0%BD%D0%B0%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM.pdf)