Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Жарикова Таисия Александровна

Содержание

1 Цель работы ........................................................................................................………………………………...3

2 Задание ..............................................................................................................…………………………………... 4

3 Теоретическое введение ......................................................................………………………….............. 5

4 Выполнение лабораторной работы

4.1 Создание программы Hello world!.............................................…………………….........................7

4.2 Работа с транслятором NASM............................................................………………………................8

4.3 Работа с расширенным синтаксисом командной строки NASM…..………..…….9

4.4 Работа с компоновщиком LD..........................................................……………………......................9

4.5 Запуск исполняемого файла.......................................................................…………………….........9

4.6 Выполнение заданий для самостоятельной работы.......................……….................10

5 Выводы ..........................................................................................…………………………………......................12

6 Источники .......................................................................................……………………………….....................13

# 1 Цель работы

Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 2 Задание

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла
6. Выполнение заданий для самостоятельной работы.

# 3 Теоретическое введение

Основными компонентами любой ЭВМ (электронно-вычислительной машины) являются центральный процессор, память и периферийные устройства. Эти элементы взаимодействуют через общую шину, которая физически представляет собой набор проводников, соединяющих устройства. В современных компьютерах проводники выполнены в виде дорожек на материнской плате. Основная задача процессора заключается в обработке информации и координации всех компонентов системы. Центральный процессор состоит из следующих элементов: 1. Арифметико-логическое устройство (АЛУ) — выполняет арифметические и логические операции для обработки данных, хранящихся в памяти. 2. Устройство управления (УУ) — управляет работой всех компонентов компьютера. 3. Регистры — быстрая внутренняя память процессора для хранения промежуточных результатов вычислений. Регистры делятся на два типа: общего назначения и специальные. Для программирования на языке ассемблера важно знать, какие регистры доступны и как они используются, поскольку большинство команд работают с данными, находящимися в регистрах. В архитектуре x86 регистры имеют следующие обозначения: - 64-битные: RAX, RCX, RDX, RBX, RSI, RDI - 32-битные: EAX, ECX, EDX, EBX, ESI, EDI - 16-битные: AX, CX, DX, BX, SI, DI - 8-битные: AH, AL, CH, CL, DH, DL, BH, BL Ещё одним важным компонентом ЭВМ является оперативная память (ОЗУ), которая представляет собой быстродействующее энергозависимое устройство, предназначенное для хранения программ и данных, с которыми процессор работает в данный момент. ОЗУ состоит из ячеек памяти, каждая из которых имеет свой адрес. Периферийные устройства включают: - Устройства внешней памяти для долгосрочного хранения данных. - Устройства ввода-вывода для взаимодействия компьютера с внешней средой. Компьютер работает на основе принципа программного управления, выполняя задачи как последовательность действий, записанных в программе. Команды процессора имеют двоичное представление и состоят из 5операционной и адресной частей. Операционная часть хранит код команды, а адресная — данные или их адреса. Командный цикл процессора включает следующие этапы: 1. Формирование адреса команды в памяти. 2. Чтение команды и её декодирование. 3. Выполнение команды. 4. Переход к следующей команде. Язык ассемблера (asm) — это машинно-ориентированный язык низкого уровня. NASM — это популярный ассемблер, поддерживающий синтаксис Intel и инструкции архитектуры x86-64, позволяющий создавать объектные файлы для различных операционных систем.

# 4 Выполнение лабораторной работы

**4.1 Создание программы Hello world!** С помощью утилиты cd перемещаюсь в каталог, в котором буду работать. Создаю в текущем каталоге пустой текстовый файл hello.asm с помощью утилиты touch (рис. 1).

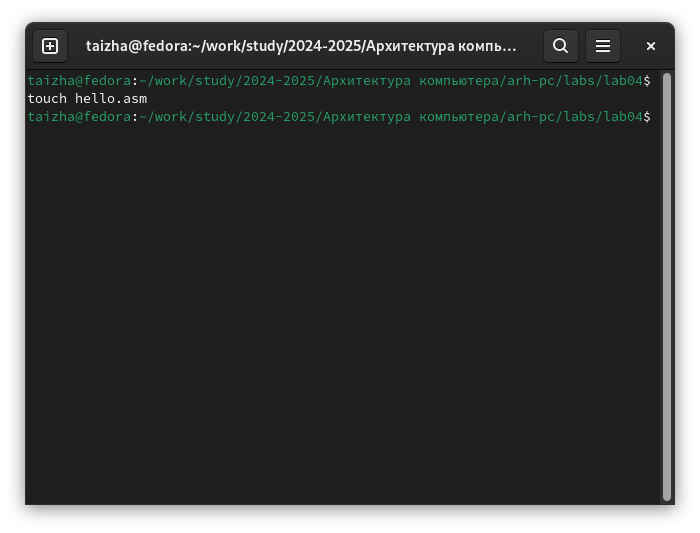


Рис. 1: 1

Открываю созданный файл в mousepad (рис. 2).

Рис. 2: 2

Заполняю файл, вставляя в него программу для вывода “Hello word!” (рис. 3).

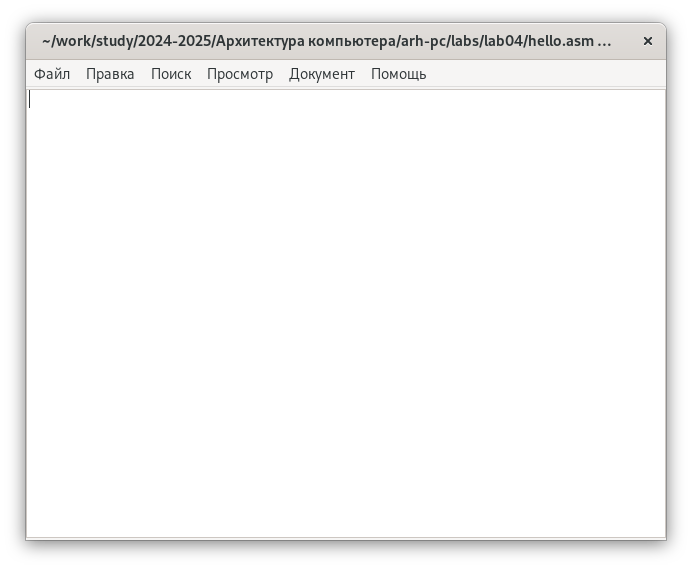


Рис. 3: 3

**4.2 Работа с транслятором NASM** Я трансформирую текст программы, выводящей “Hello, world!”, в объектный код с помощью транслятора NASM, используя команду nasm -f elf hello.asm. Ключ -f указывает NASM на необходимость создания бинарного файла в формате elf. Затем с помощью утилиты ls проверяю, что команда выполнена корректно — действительно, создан файл “hello.o” (рис. 4).

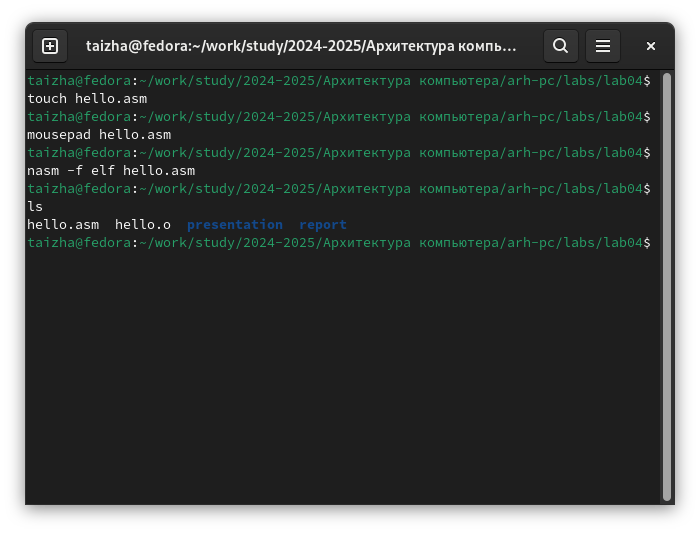


Рис. 4: 4

**4.3 Работа с расширенным синтаксисом командной строки NASM** Я ввожу команду для компиляции файла hello.asm в объектный файл obj.o, используя ключ -g для включения отладочных символов. Также с помощью ключа -l создается файл листинга list.lst. Затем с помощью утилиты ls проверяю корректность выполнения команды (рис. 5).

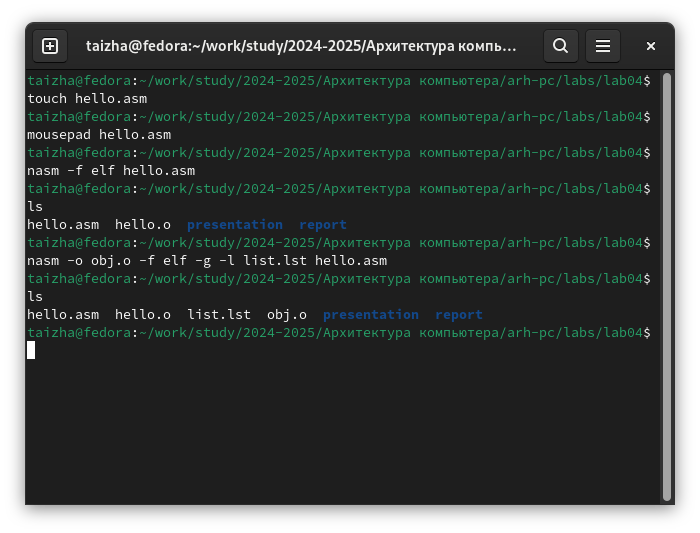


Рис. 5: 5

**4.4 Работа с компоновщиком LD** Я передаю объектный файл hello.o на обработку компоновщику LD для создания исполняемого файла hello. Ключ -o указывает имя создаваемого исполняемого файла. После этого с помощью утилиты ls проверяю корректность выполнения команды (рис. 6).

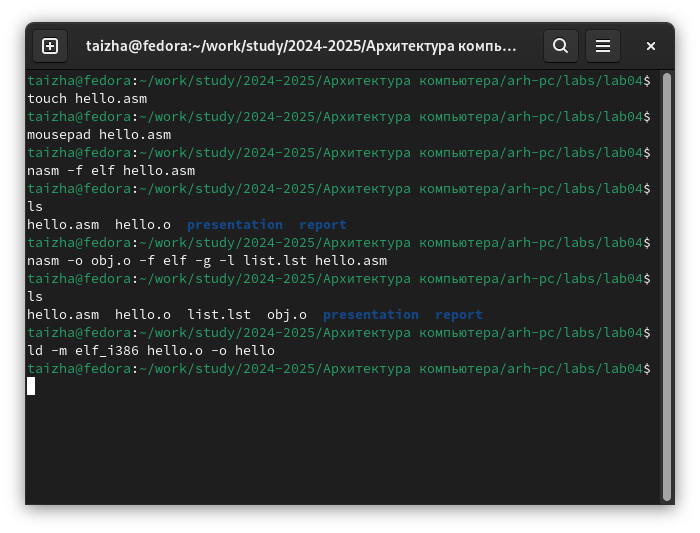


Рис. 6: 6

Я выполняю следующую команду, в результате которой исполняемый файл получит имя main, так как после ключа -o указано значение main. Объектный файл, использованный для создания этого исполняемого файла, называется obj.o (рис. 7).

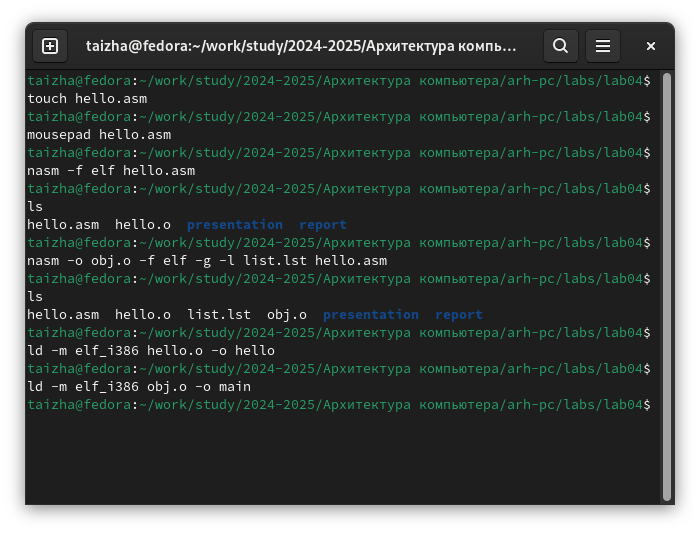


Рис. 7: 7

**4.5 Запуск исполняемого файла** Запускаю на выполнение созданный исполняемый файл hello (рис. 8).

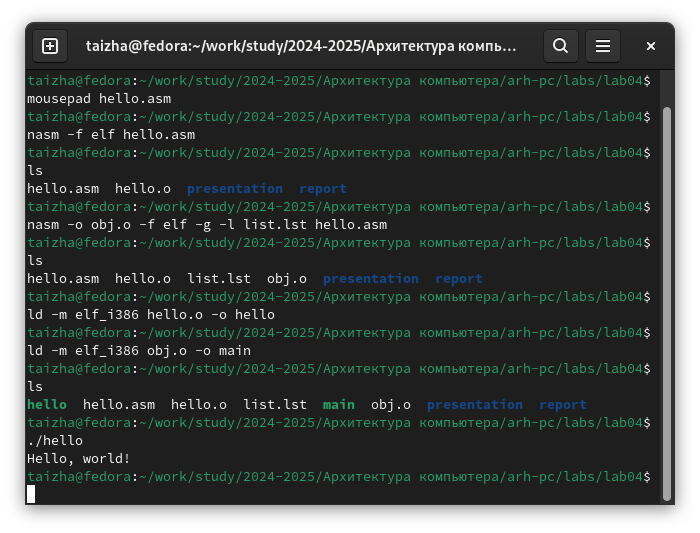


Рис. 8: 8

**4.6 Выполнение заданий для самостоятельной работы** С помощью утилиты cp создаю в текущем каталоге копию файла hello.asm с именем lab04.asm и проверяю правильность (рис. 9).

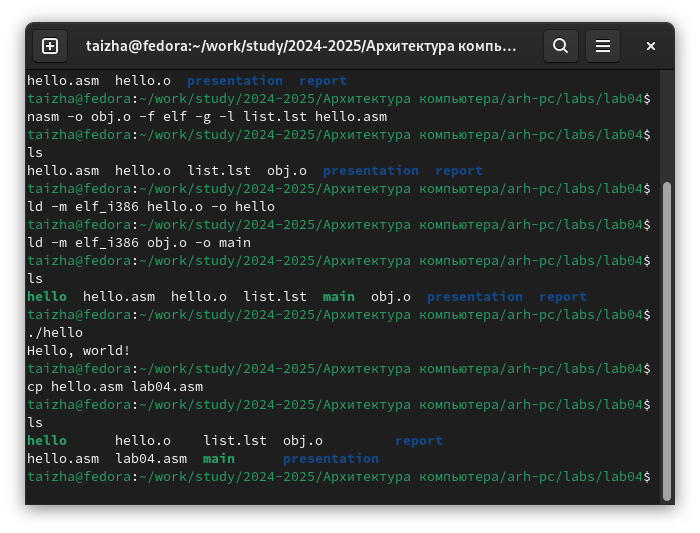


Рис. 9: 9

С помощью текстового редактора mousepad открываю файл lab04.asm и вношу изменения в программу так, чтобы она выводила мои имя и фамилию (рис. 10).

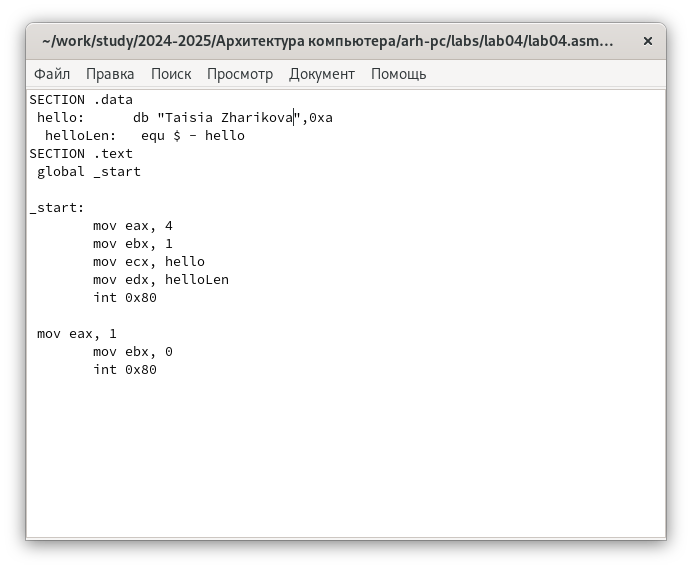


Рис. 10: 10

Компилирую текст программы в объектный файл. Проверяю с помощью утилиты ls, что файл lab04.o создан (рис. 11).

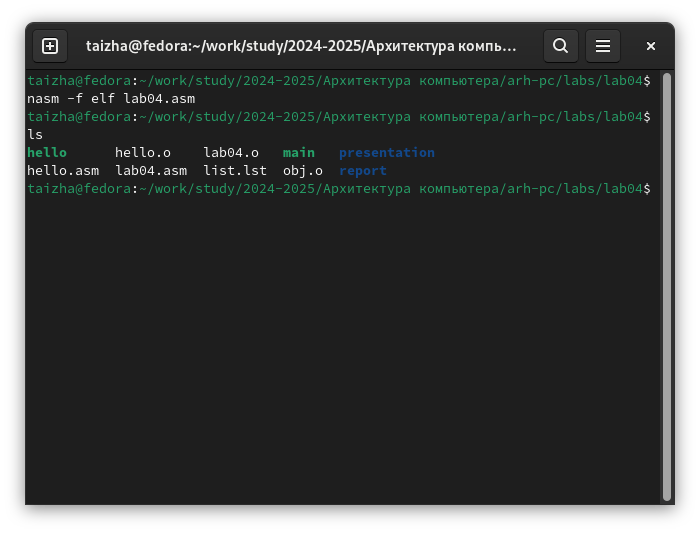


Рис. 11: 11

Передаю объектный файл lab04.o на обработку компоновщику LD, чтобы получить исполняемый файл lab04 (рис. 12).

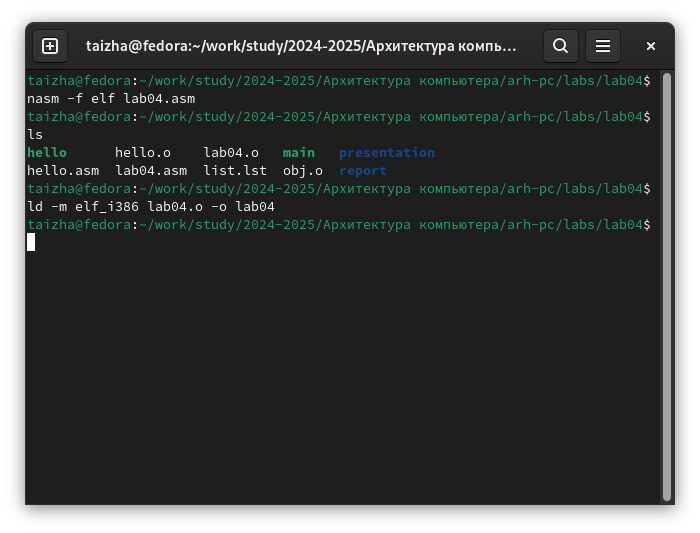


Рис. 12: 12

Запускаю исполняемый файл lab04, на экран действительно выводятся мои имя и фамилия (рис. 13).

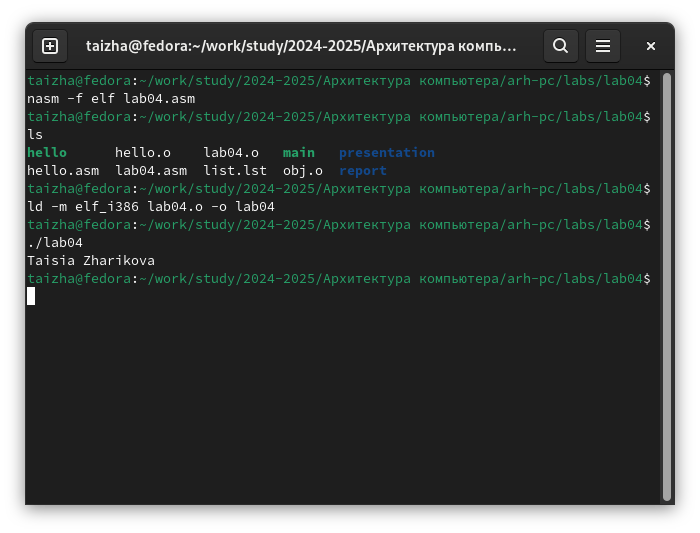


Рис. 13: 13

С помощью команд git add . и git commit добавляю файлы на GitHub, комментируя действие как добавление файлов для лабораторной работы №4 (рис. 14).

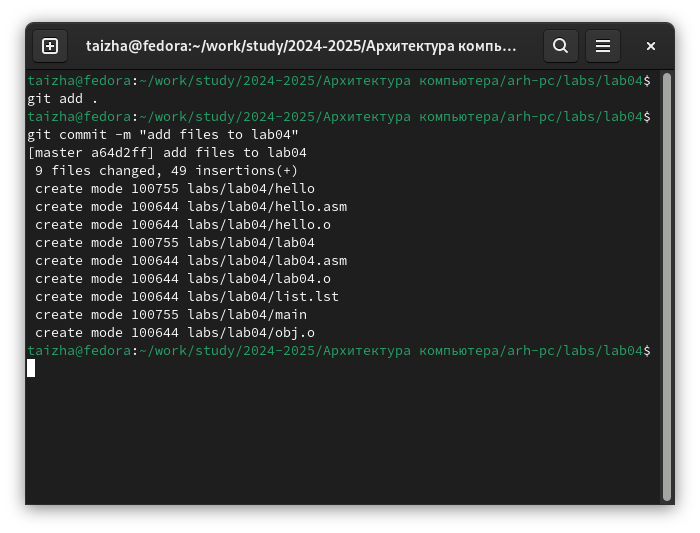


Рис. 14: 14

Отправляю файлы на сервер с помощью команды git push (рис. 15).

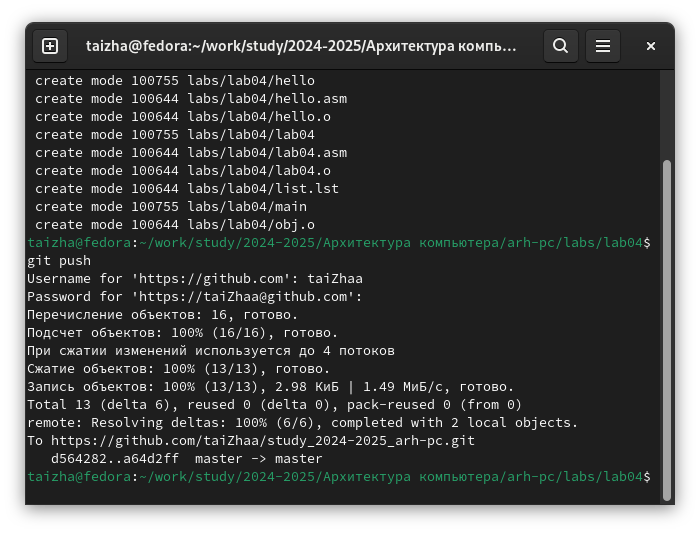


Рис. 15: 15

# 5 Выводы

При выполнении данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 6 Список литературы

1. [Архитектура ЭВМ (rudn.ru)](https://esystem.rudn.ru/)