Отчёт по лабораторной работе №4

Архитектура компьютера

Баптишта Матеуж , НКАбд-01-23

Содержание

# 1 Цель работы

В пятой лабораторной работе мы рассмотрим, как освоить процедуру компиляции и сборки программ, написанных на ассемблере nasm.

# 2 Выполнение лабораторной работы

**Программа Hello world!** В этом разделе мы хотели создать программу,которая выводит строку “Hello world!” но на языке ассемблера nasm. Вот почему мы начали с рекурсивного создания нового каталога “~/work/arch-pc/lab05”. (рис. [1](#fig:1))

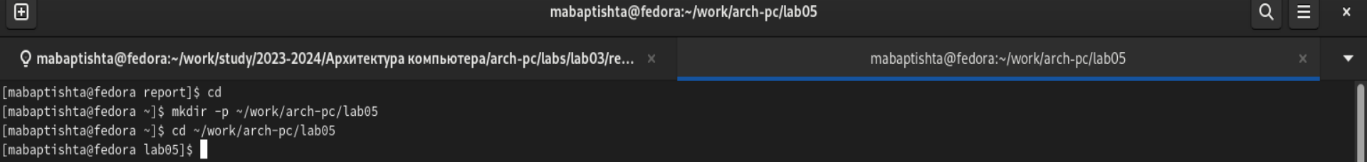


Figure 1: Рисунок 1

После этого мы создали текстовый файл в формате .asm, затем открываем только что созданный файл с помощью текстового редактора gedit. (рис. [2](#fig:2))

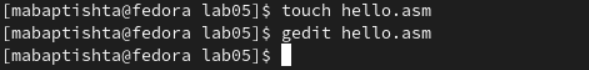


Figure 2: Рисунок 2

После этого мы добавили код сборки, который выводит “Hello world!” в файл hello.asm. (рис. [3](#fig:3))

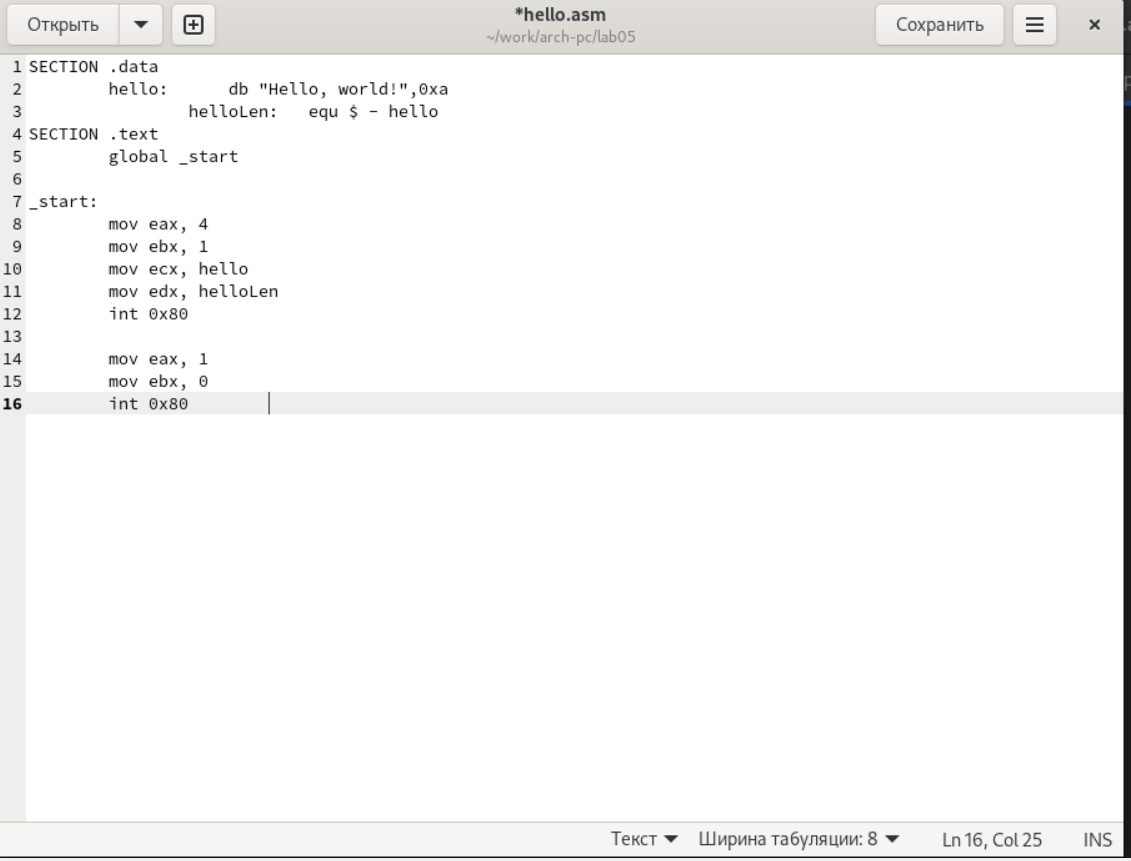


Figure 3: Рисунок 3

**Транслятор NASM**

На этом этапе,используя переводчик NASM,мысмогли скомпилировать или перевести код в объектный код, который создал другой файл с форматом .o. (рис. [4](#fig:4))

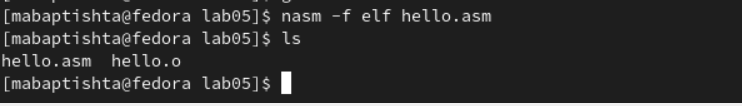


Figure 4: Рисунок 4

Используя команду ls, мы проверили работу, проделанную переводчиком, и обнаружили, что объектный файл был создан с тем же именем, что и текстовый файл.

**Расширенный синтаксис командной строки NASM**

Здесь мы запустили полную команду NASM и проверили выходные файлы, которые дала нам.Разница заключалась в том, что с помощью полной команды нам нужно указать имя объектного файла и список файлов, и это то, что получилось после проверки с помощью запятой ls. (рис. [5](#fig:5))

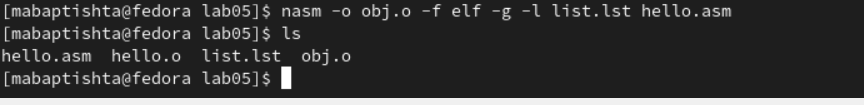


Figure 5: Рисунок 5

**Компоновщик LD**

На этом шаге и с помощью компоновщика с командой ld мы смогли получить исполняемый файл, обработав объектный файл. Затем,используя команду ls, мы проверили, что файл был создан. (рис. [6](#fig:6))

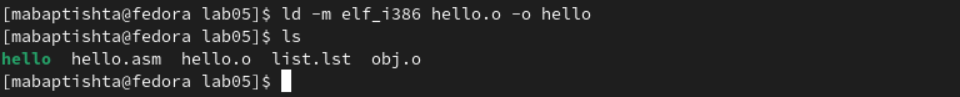


Figure 6: Рисунок 6

Затем мы проверили, что можем присвоить исполняемому файлу любое имя,анетолькотожеимя,чтоиобъектномуфайлу,какпоказаноспомощью команды ls. (рис. [7](#fig:7))

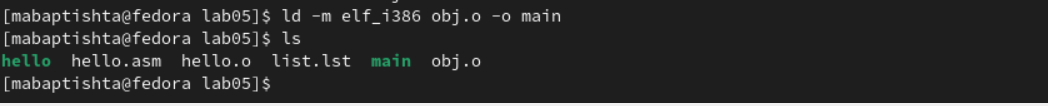


Figure 7: Рисунок 7

Исполняемый файл имеет имя “main”, а для объектного файла -“obj”.

**Запуск исполняемого файла**

На этом шаге все, что мы сделали, это запустили исполняемый файл. (рис. [8](#fig:8))



Figure 8: Рисунок 8

**Выводы по результатам выполнения заданий**

В этой лабораторной работе мы освоили, как скомпилировать текстовый файл, написанный на языке ассемблера NASM, в объектный файл, получить оправдание,и все это ради создания программы,которая печатает знаменитое предложение “Hello world!”

# 3 Задание для самостоятельной работы

1. В каталоге ~/work/arch-pc/lab05 мы создали копию для файла hello.asm и присвоили ему имя lab05. (рис. [9](#fig:9))

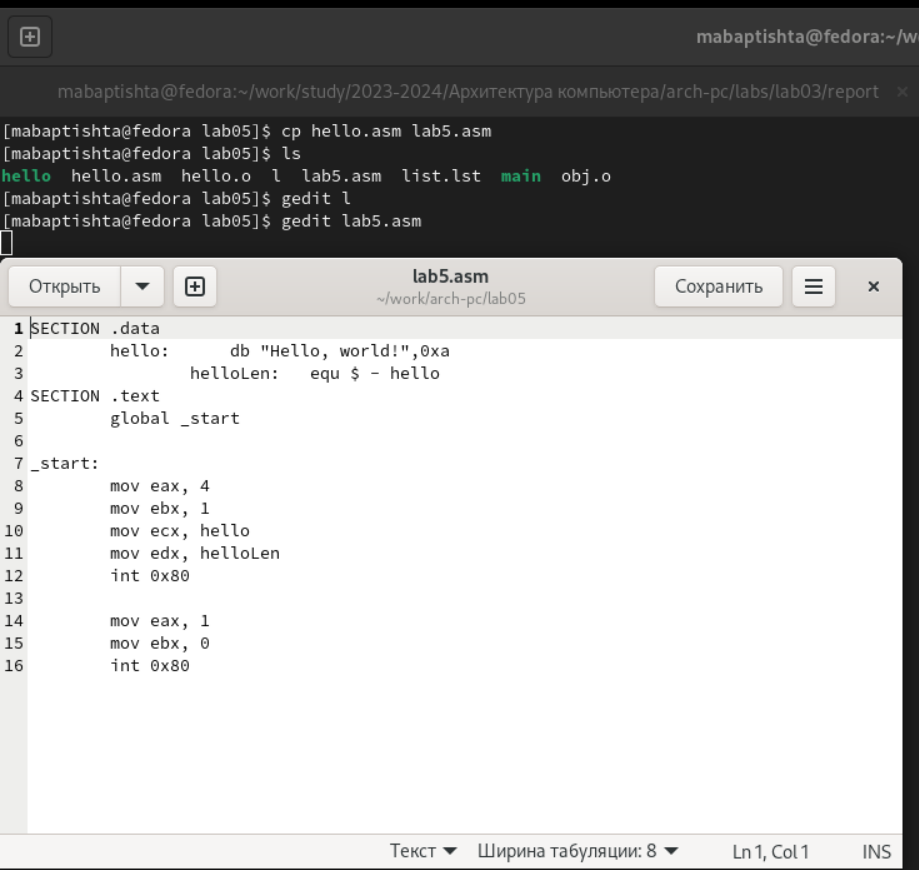


Figure 9: Рисунок 9

1. Используя текстовый редактор gedit, мы изменили текстовый файл, содержащий ассемблерный код, чтобы программа выводила мое имя и фамилию “Baptista Mateus”.

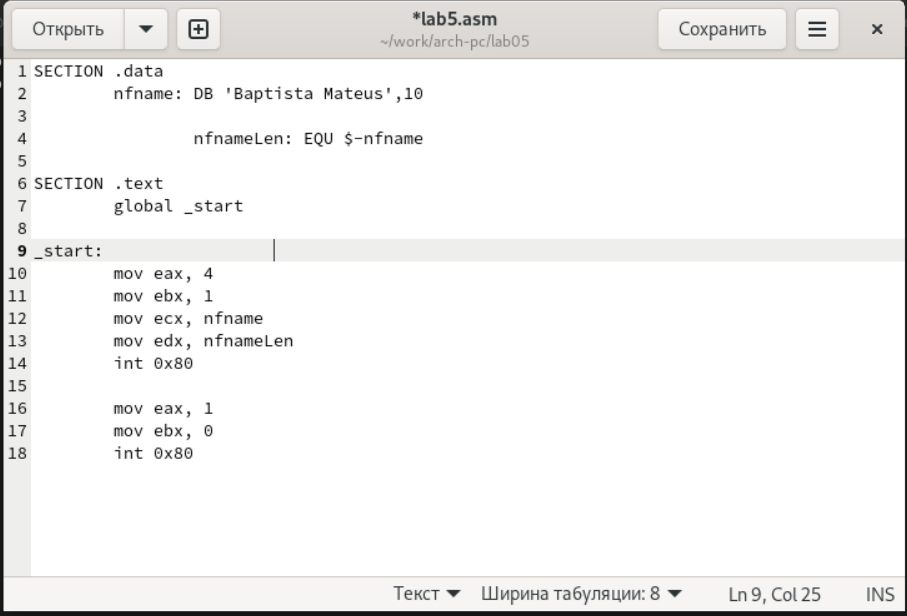


Figure 10: Рисунок 10

3.0.1 листинг написанной программы: SECTION .data nfname: DB ‘Baptista Mateus’,10 nfnameLen: EQU $-nfname SECTION .text global \_start \_start: mov eax, 4 mov ebx, 1 mov ecx, nfname mov edx, nfnameLen int 0x80 mov eax, 1 mov ebx, 0 int 0x80

1. После написания кода e скомпилировал код в объектный файл после чего получил исполняемый файл с помощью компоновщика. (рис. [11](#fig:11))

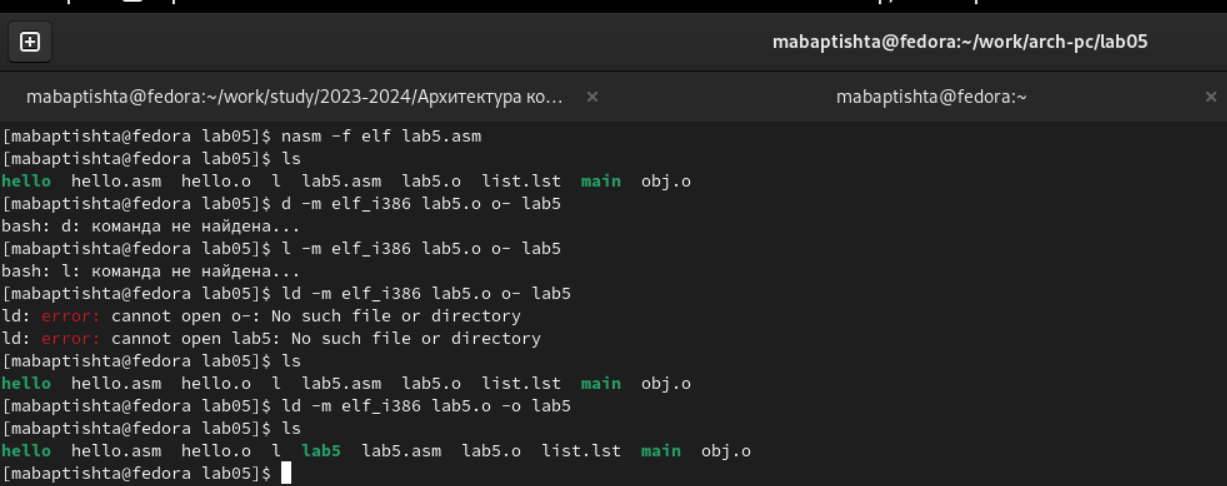


Figure 11: Рисунок 11

Затем мы запустили исполняемый файл. (рис. [12](#fig:12))

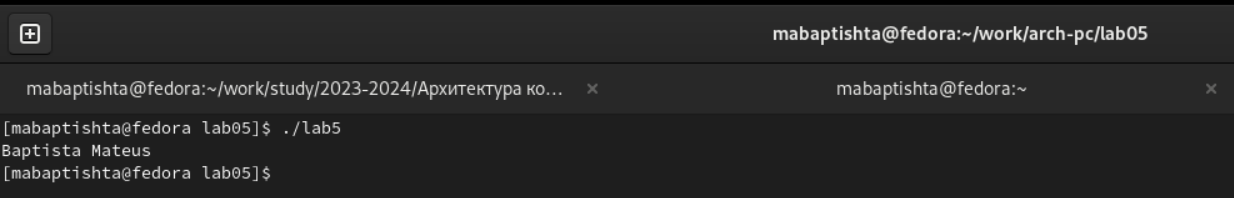


Figure 12: Рисунок 12

1. Здесь мы скопировали оба hello.Asm и lab5.asm в ваш локальный репозиторий. (рис. [13](#fig:13))

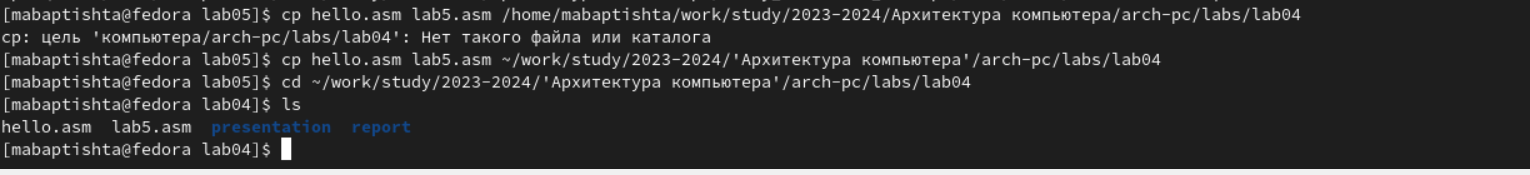


Figure 13: Рисунок 13

Наконец, мы загрузили все файлы в удаленный репозиторий. (рис. [14](#fig:14))

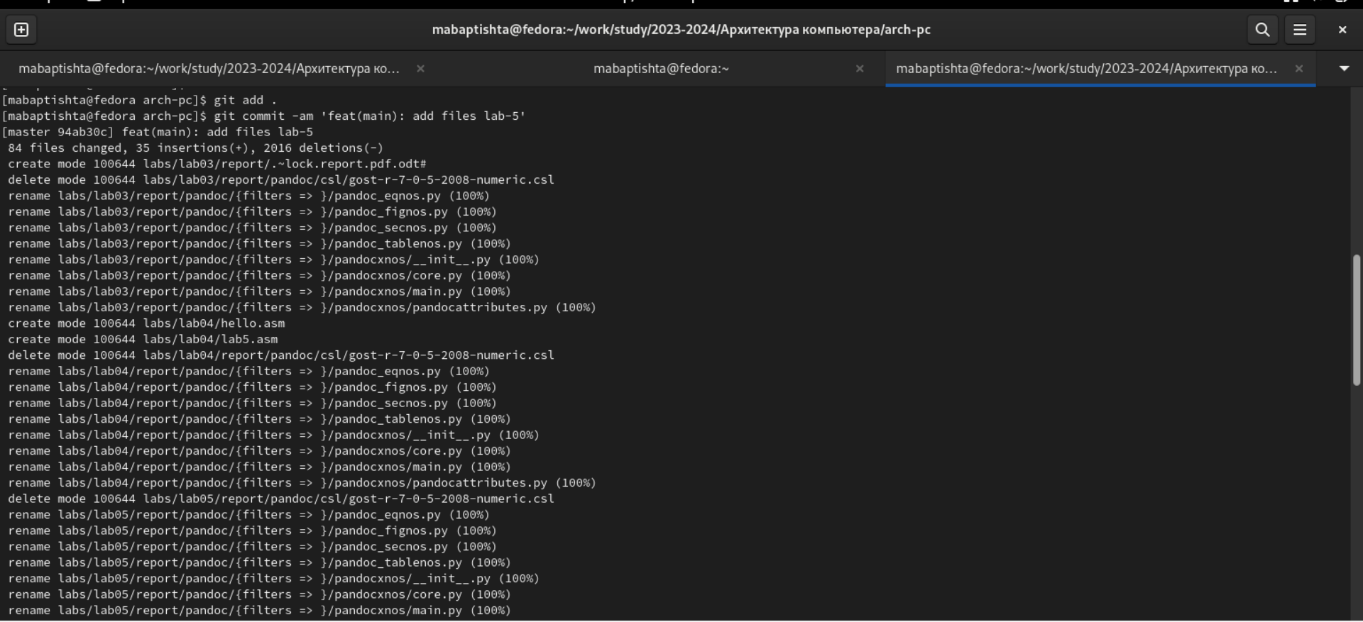


Figure 14: Рисунок 14

**Выводы по результатам выполнения заданий** В этих упражнениях мы применили навыки, полученные в ходе лабораторной работы, в ходе которой получили более глубокое представление об именах регистров и о том, как выделить для них память.

# 4 Выводы

В шестой лабораторной работе мы можем получить практические навыки по созданию компиляции и обработке программы с использованием языка ассемблера Nasm