Отчет по лабораторной работе №4

Дисциплина: Архитектура компьютера

Мустафина Аделя Юрисовна

Содержание

[Цель работы 1](#__RefHeading___Toc178_3075642298)

[Задание 1](#__RefHeading___Toc180_3075642298)

[Теоретическое введение 2](#__RefHeading___Toc182_3075642298)

[Выполнение лабораторной работы 4](#__RefHeading___Toc184_3075642298)

[4.3.1. Программа Hello world! 4](#__RefHeading___Toc186_3075642298)

[4.3.2. Транслятор NASM 7](#__RefHeading___Toc188_3075642298)

[4.3.3. Расширенный синтаксис командной строки NASM 7](#__RefHeading___Toc190_3075642298)

[4.4. Компоновщик LD 8](#__RefHeading___Toc192_3075642298)

[4.4.1. Запуск исполняемого файла 8](#__RefHeading___Toc194_3075642298)

[4.5. Задание для самостоятельной работы 9](#__RefHeading___Toc196_3075642298)

[Выводы 10](#__RefHeading___Toc198_3075642298)

[Список литературы 10](#__RefHeading___Toc200_3075642298)

# Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 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.

# Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины

(ЭВМ) являются центральный процессор, память и периферийные устройства (рис. 4.1).

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; регистры процессора делятся на два типа: регистры общего назначения и cпециальные регистры.

Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблере используют регистры в качестве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, преобразование (арифметические или логические операции) данных хранящихся в регистрах.

В качестве примера приведем названия основных регистров общего назначения (именно эти регистры чаще всего используются при написании программ):

- RAX, RCX, RDX, RBX, RSI, RDI — 64-битные

- EAX, ECX, EDX, EBX, ESI, EDI — 32-битные

- AX, CX, DX, BX, SI, DI — 16-битные

- AH, AL, CH, CL, DH, DL, BH, BL — 8-битные

Другим важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ). ОЗУ — это быстродействующее энергозависимое запоминающее устройство, которое напрямую взаимодействует с узлами процессора, предназначенное для хранения программ и данных, с которыми процессор непосредственно работает в текущий момент. ОЗУ состоит из одинаковых пронумерованных ячеек памяти. Номер ячейки памяти — это адрес хранящихся в ней данных.

Периферийные устройства в составе ЭВМ:

- устройства внешней памяти, которые предназначены для долговременного хранения больших объёмов данных.

- устройства ввода-вывода, которые обеспечивают взаимодействие ЦП с внешней средой.

В основе вычислительного процесса ЭВМ лежит принцип программного управления. Это означает, что компьютер решает поставленную задачу как последовательность действий, записанных в виде программы.

Коды команд представляют собой многоразрядные двоичные комбинации из 0 и 1. В коде машинной команды можно выделить две части: операционную и адресную. В операционной части хранится код команды, которую необходимо выполнить. В адресной части хранятся данные или адреса данных, которые участвуют в выполнении данной операции.

При выполнении каждой команды процессор выполняет определённую последовательность стандартных действий, которая называется командным циклом процессора. Он заключается в следующем:

1. формирование адреса в памяти очередной команды;

2. считывание кода команды из памяти и её дешифрация;

3. выполнение команды;

4. переход к следующей команде.

Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня.

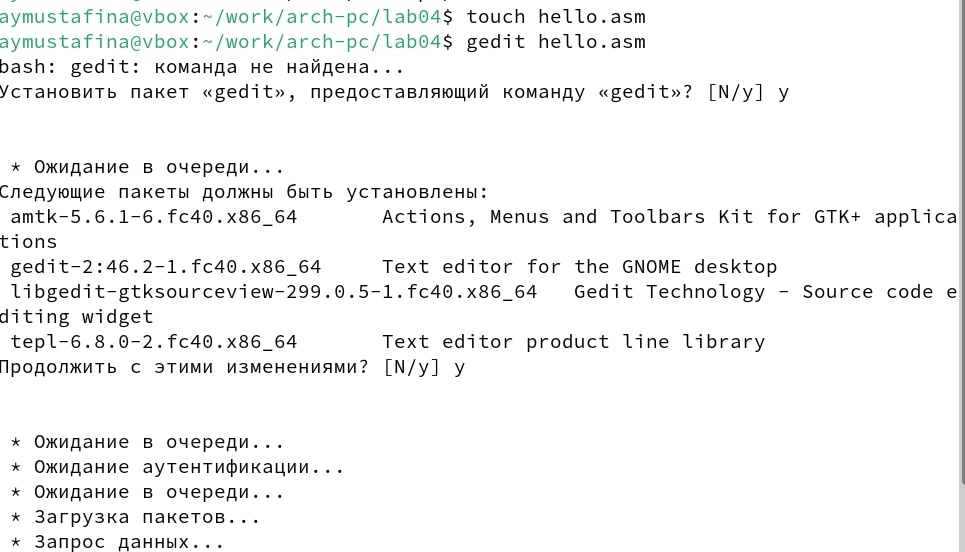
NASM — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64.

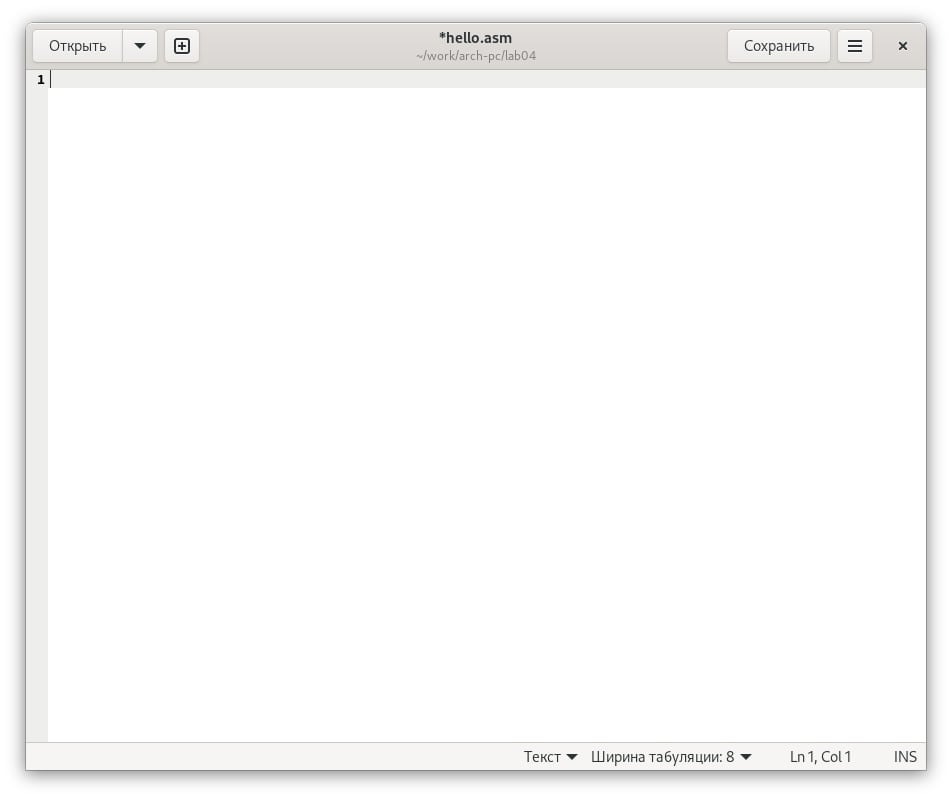
# Выполнение лабораторной работы

## 4.3.1. Программа Hello world!

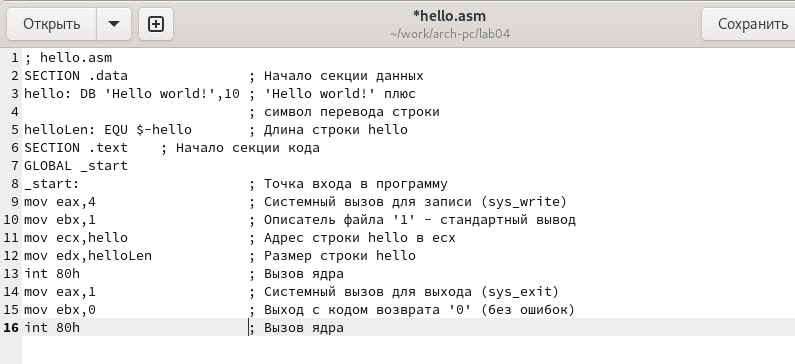
Перемещаюсь по каталогам для создания каталога для работы с программами на языке ассемблера. Но далее я скопировала все в другой каталог для своего удобства.

Создаю текстовый файл с именем hello.asm с помощью команды touch. И открываю этот файл с помощью текстового редактора gedit. Дополнительно его загружаю, так как на моей fedora его нет.



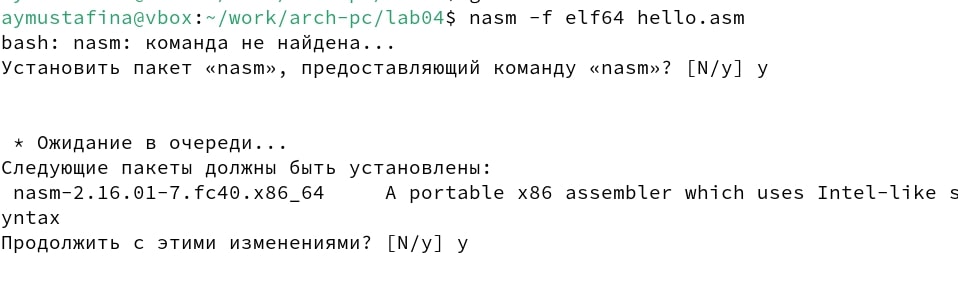


Открытый исходный файл

Ввожу в данный файл код для вывода “Hello world!”.

## 4.3.2. Транслятор NASM

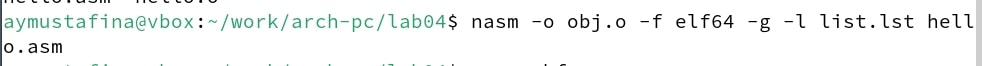
Теперь для того, чтобы скомпилировать данный текст программы использую транслятор NASM.Я снова загружаю недостающую команду.

После загрузки снова ввожу эту команды и проверяю скомпилировался ли необходимый мне файл. Объектный файл имеет имя “hello.0”.

## 4.3.3. Расширенный синтаксис командной строки NASM

Теперь с помощью другой команды компилирую исходный файл hello.asm в obj.o, с ним же будет создан файл листинга list.lst. Снова проверяю правильность выполнения с помощью команды ls.



Компиляция файлов(2)

На этом моменте я перехожу в другой каталог, скопировав все созданные файлы для своего удобства.

## 4.4. Компоновщик LD

Объектный файл передам на обработку компоновщику и проверю правильность выполнения команды.

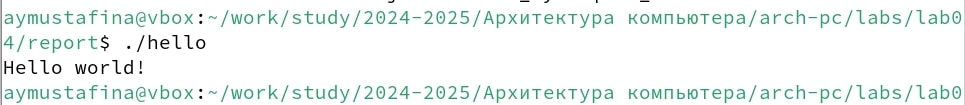


Далее выполняю еще одну команду. Исполняемый файл имеет имя “main”.



## 4.4.1. Запуск исполняемого файла

Запускаю созданный исполняемый файл, набрав в командной строке ./hello.

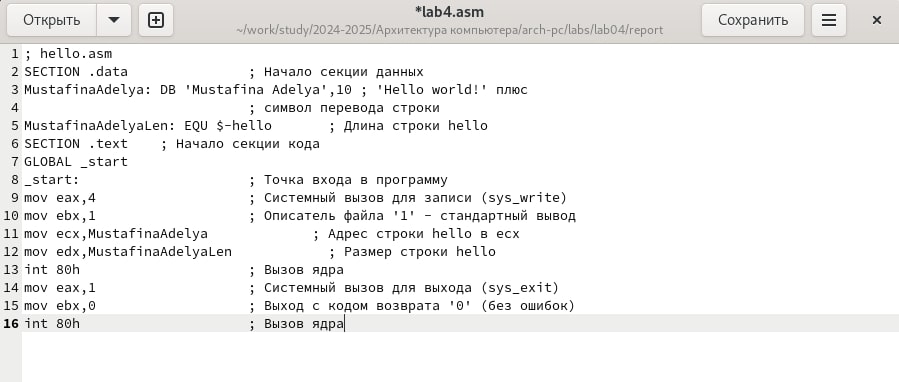
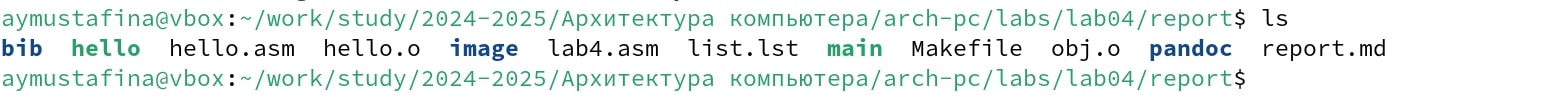


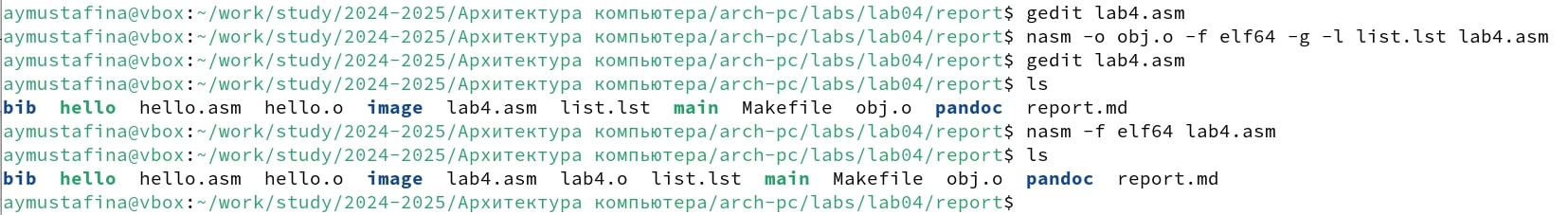
## 4.5. Задание для самостоятельной работы

Создаю копию файла hello.asm с именем lab4.asm с помощью команды cp.



С помощью текстового редактора вношу изменения в файл, чтобы он выводил на экран мое имя и фамилию.



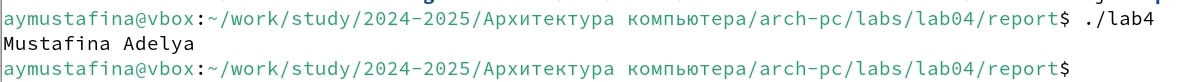
Выполняю те же команды как для файла с hello.

Преобразование файлов



Преобразование файлов

Вывожу свое имя и фамилию.



# Выводы

При выполнении лабораторной работы я изучила компиляцию файлов написанных на ассемблере NASM.  
Здесь кратко описываются итоги проделанной работы.

# Список литературы

1. https://esystem.rudn.ru/pluginfile.php/2089084/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