**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**Лабораторная работа №4**

**на тему «Создание и процесс обработки программ на языке ассемблера NASM»**

*дисциплина: Архитектура компьютера*

Студент: Трусова А.А.

Группа: НКАбд-05-24

№ ст. билета: 1132246715

**МОСКВА**

2024г.

**Содержание  
Цель работы**

**Теоретическое введение**

**Выполнение лабораторной работы**

**4.1.**  Программа Hello world!

**4.2.** Транслятор NASM

**4.3.** Расширенный синтаксис командной строки NASM

**4.4.** Компоновщик LD

**4.5.** Запуск исполняемого файла

**4.6.** Задание для самостоятельной работы

**Вывод**

**Цель работы**

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

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

Доступ к регистрам осуществляется не по адресам, как к основной памяти, а по именам. Каждый регистр процессора архитектуры x86 имеет свое название, состоящее из 2 или 3 букв латинского алфавита.

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 (половинки 16-битных регистров).

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.

В состав ЭВМ также входят периферийные устройства, которые можно разделить на:

* устройства внешней памяти, которые предназначены для долговременного хранения больших объёмов данных (жёсткие диски, твердотельные накопители, магнитные ленты);

• 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.

При выполнении каждой команды процессор выполняет определённую последовательность стандартных действий, которая называется командным циклом процессора. В самом общем виде 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**

**4.1. Программа Hello world!**

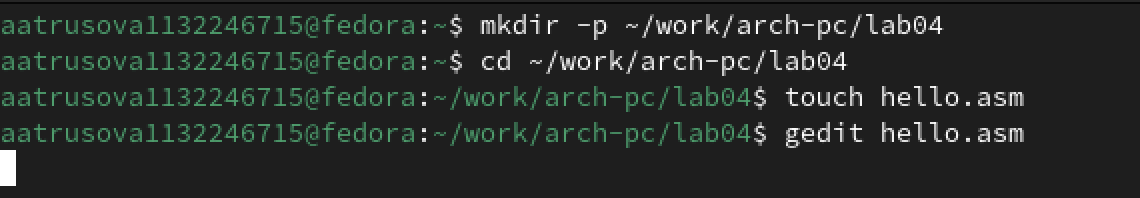
Создала каталог для работы с программами на языке ассемблера NASM, перешла в него, создала файл hello.asm и открыла его с помощью gedit (рис.1).

Рис.1

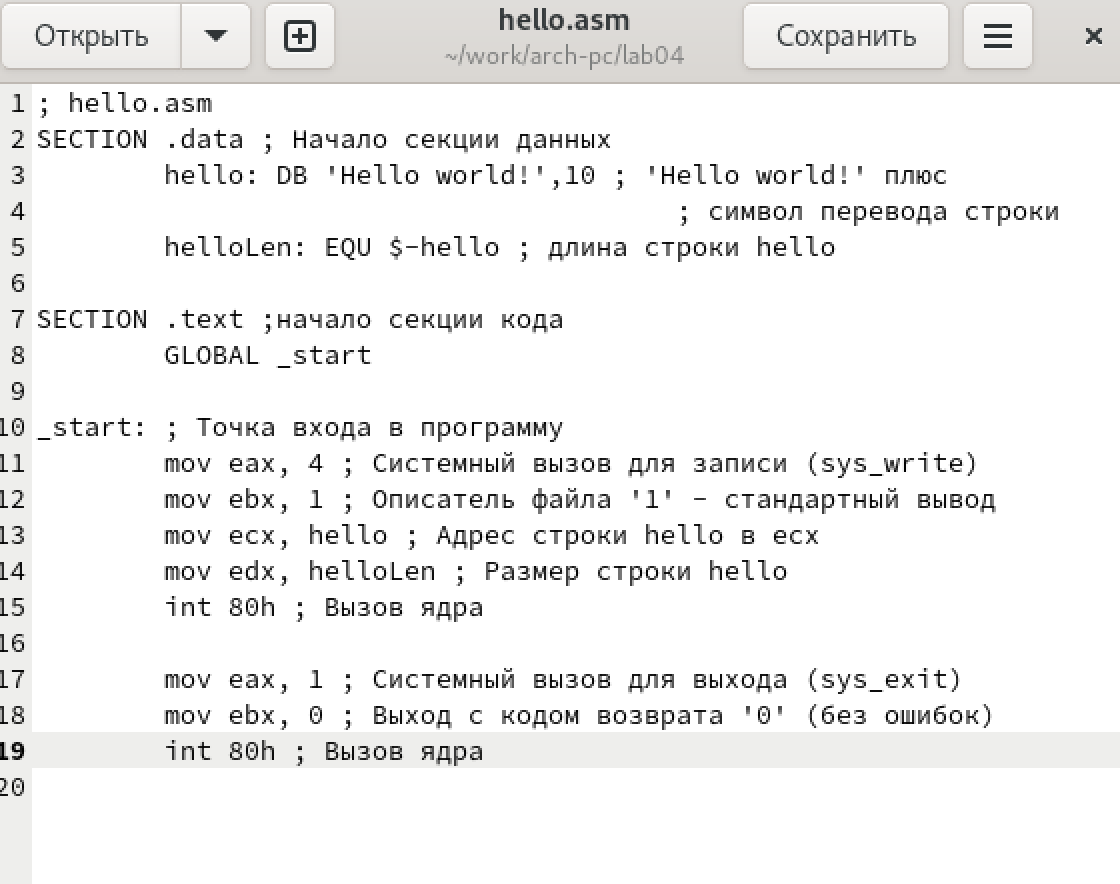
Записала в него текст программы (рис.2).

Рис.2

**4.2. Транслятор NASM**

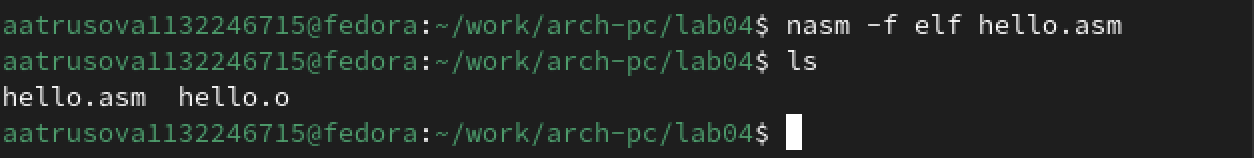
С помощью NASM преобразовала текст программы из файла hello.asm в объектный код в файл hello.o и проверила правильность выполнения(рис.3).

Рис.3

**4.3. Расширенный синтаксис командной строки NASM**

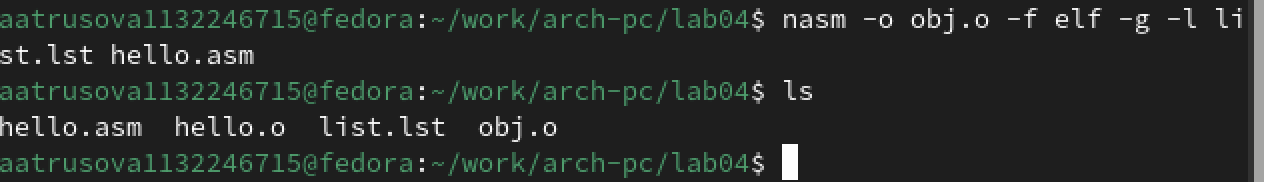
Скомпилировала исходный файл hello.asm в obj.o с форматом выходного файла elf, создала файл листинга list.lst и проверила правильность выполнения (рис.4).

Рис.4

**4.4. Компоновщик LD**

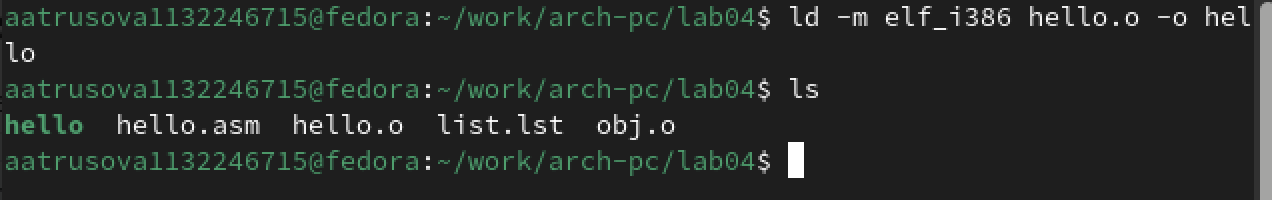
Передаю объектный файл hello.o компоновщику ld на обработку и получаю исполняемый файл hello (рис.5).

Рис.5

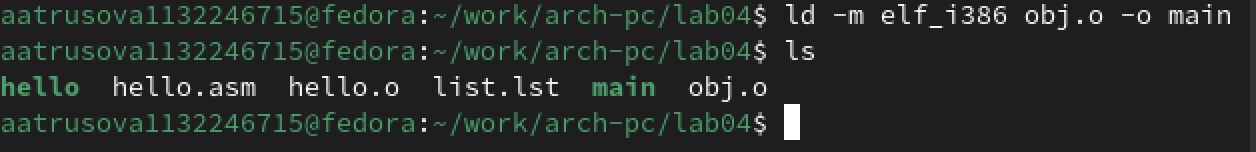
Передаю объектный файл obj.o компоновщику ld на обработку и получаю исполняемый файл main (рис.6).

Рис.6

**4.5. Запуск исполняемого файла**

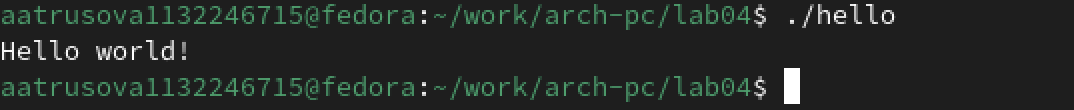
Запустила исполняемый файл. Работает корректно (рис. 7).

Рис.7

**4.6. Задание для самостоятельной работы**

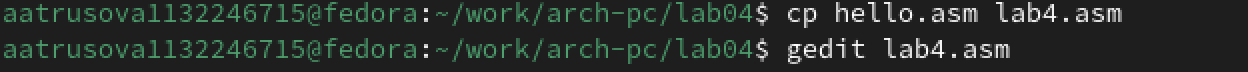
В каталоге ~/work/arch-pc/lab04 с помощью команды cp создала копию файла hello.asm с именем lab4.asm и открыла её (рис.8).

Рис.8

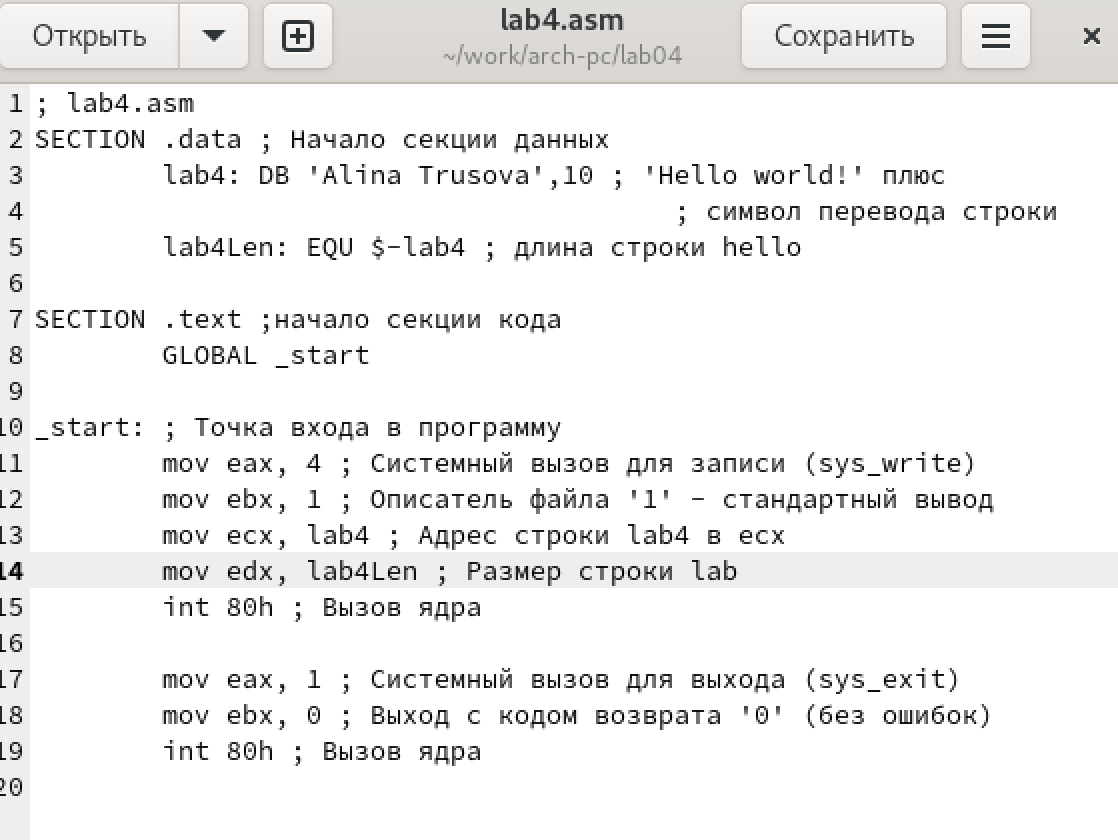
Переписала копию так, чтобы вместо «Hello world!» программа выводила «Alina Trusova» (рис.9).

Рис.9

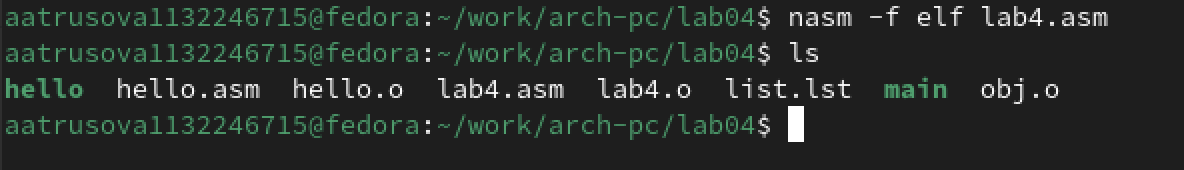
Оттранслировала файл lab4.asm в объектный файл lab4.o (рис. 10).

Рис.10

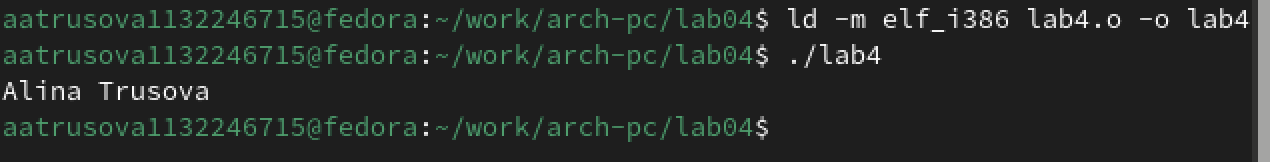
Скомпоновала объектный файл и получила исполняемый файл lab4 (рис.11).

Рис.11

Скопировала файлы hello.asm и lab4.asm в локальный репозиторий в ката-

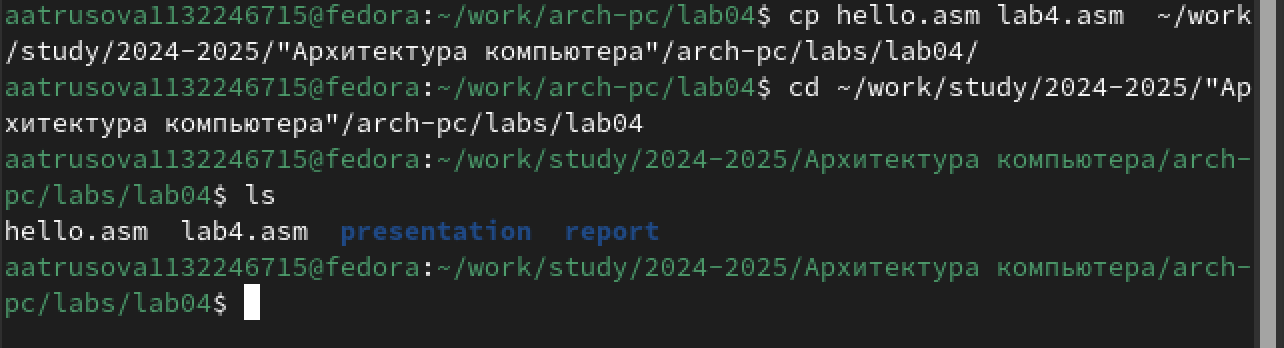
лог ~/work/study/2023-2024/"Архитектура компьютера»/arch-pc/labs/lab04/ (рис.12).

Рис.12

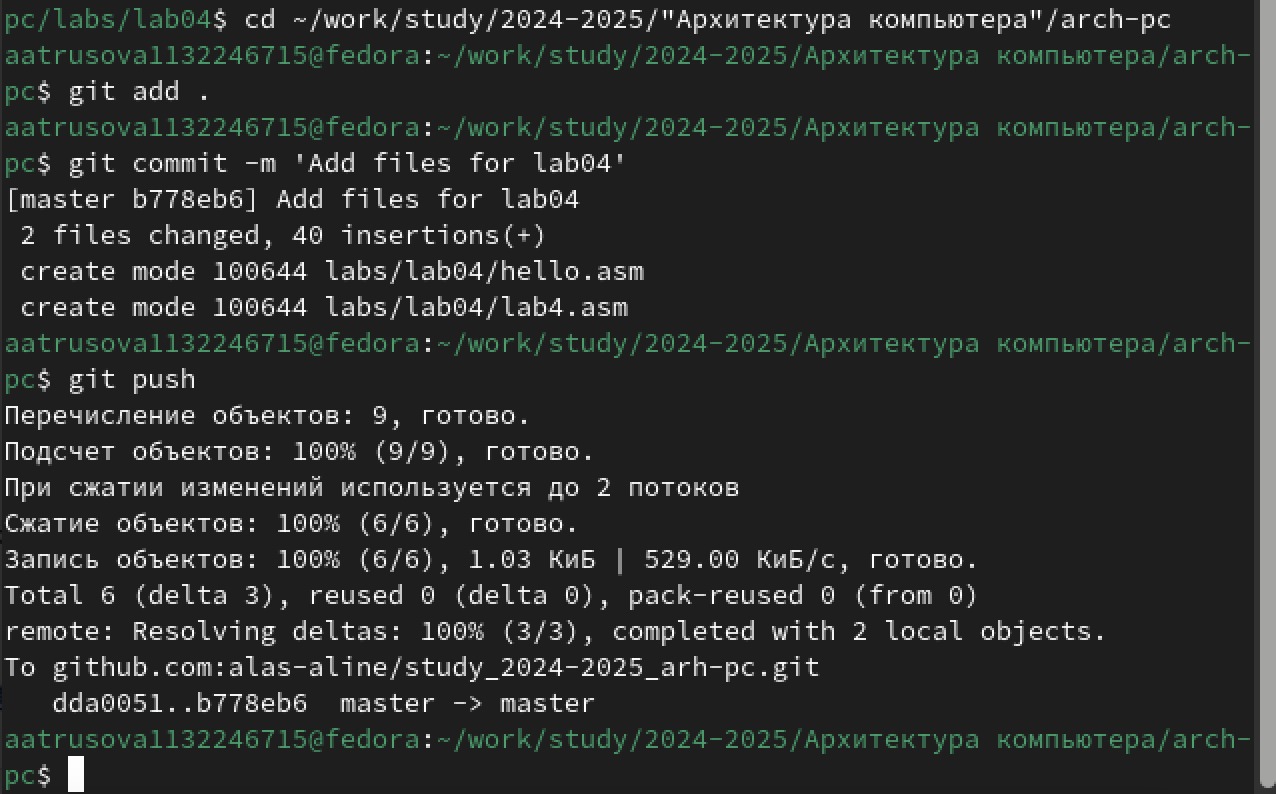
Выгрузила изменения на GitHub (рис.13).

Рис.13

**Вывод**

Я освоила процедуру компиляции и сборки программ, написанных на ассемблере NASM.

**Список литературы**

**1. https://esystem.rudn.ru/pluginfile.php/2089084/mod\_resource/content/0/Лабораторная работа №4. Создание и процесс обработки программ на языке ассемблера NASM.pdf**