Отчёт по лабораторной работе №4

Дисциплина:Архитектура компьютера

Зайцева Ульяна Владимировна

Содержание

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

# 2 Задание

1. Программа Hello world!
2. Транслятор NASM
3. Расширенный синтаксис командной строки NASM
4. Компоновщик LD
5. Запуск исполняемого файла
6. Задание для самостоятельной работы

# 3 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 специальные регистры.

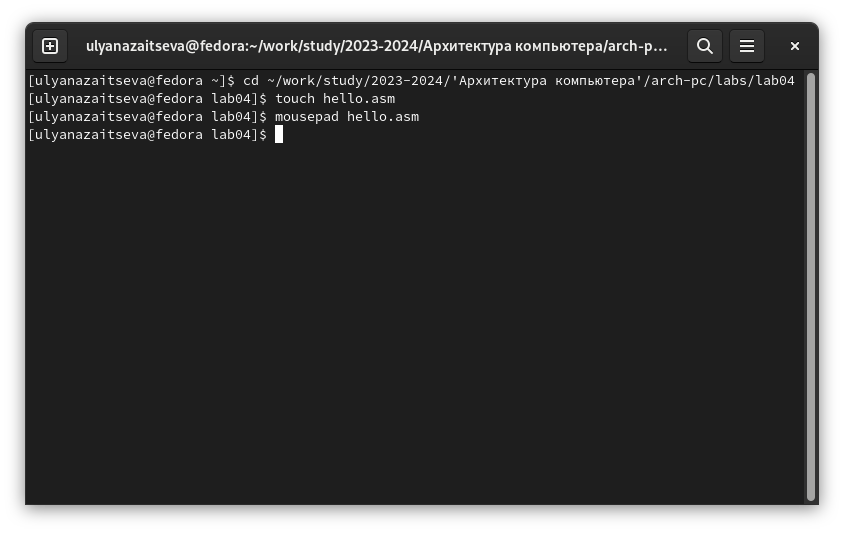
Доступ к регистрам осуществляется не по адресам, как к основной памяти, а по именам. Каждый регистр процессора архитектуры x86 имеет свое название, состоящее из 2 или 3 букв латинского алфавита. В качестве примера приведем названия основных регистров общего назначения (именно эти регистры чаще всего 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). Например, AH (high AX) — старшие 8 бит регистра AX, AL (low AX) — младшие 8 бит регистра AX.

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. Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — машинные коды.

# 4 Выполнение лабораторной работы

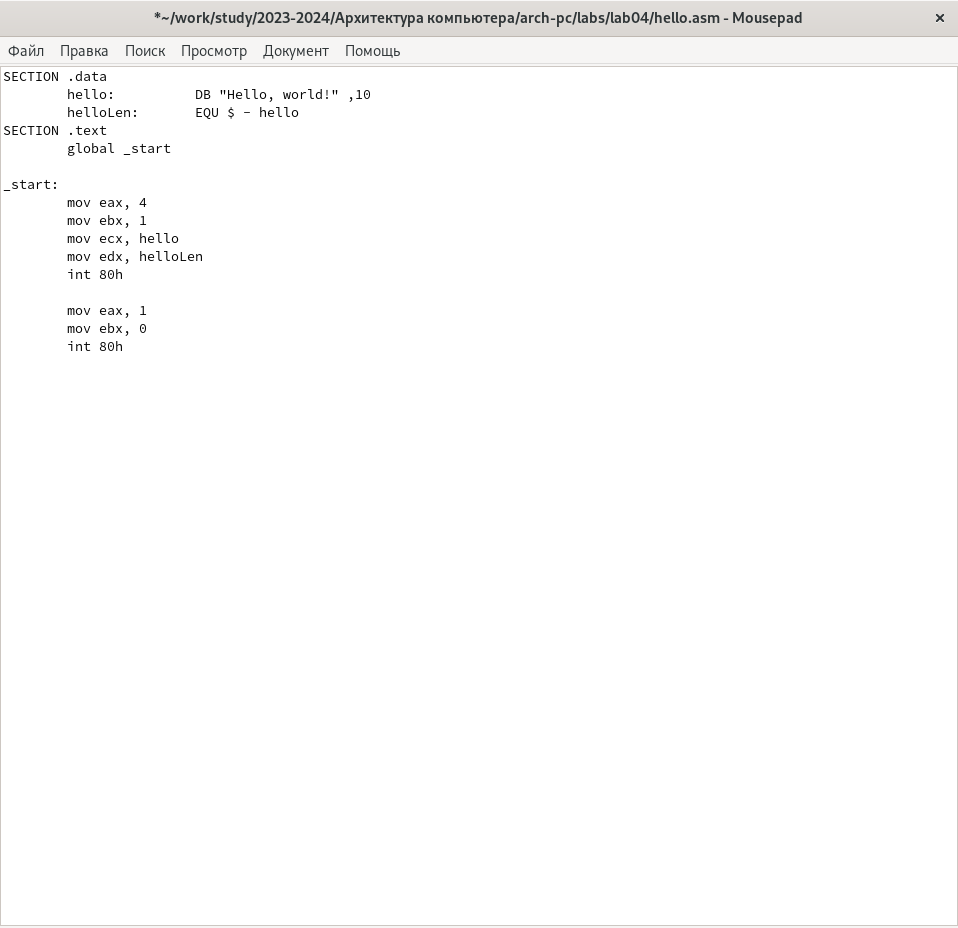
1. Программа Hello world!

Создаю каталог для работы с программами на языке ассемблера NASM и перехожу в созданный каталог.(рис. ??)



Новый каталог

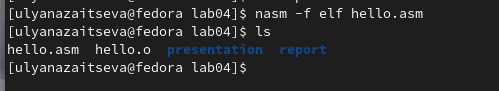
Создаю текстовый файл с именем hello.asm, открываю этот файл с помощью текстового редактора и ввожу следующий текст из файла.(рис. ??)



Текст программы

1. Транслятор NASM

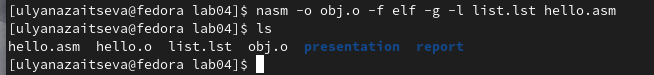
Компилирую программу с помощи команды nasm -f elf hello.asm и проверим, что файл создан.(рис. ??)



Компиляция программы

1. Расширенный синтаксис командной строки NASM

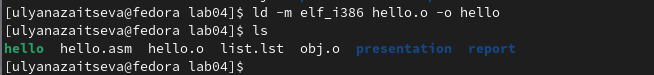
С помощью команды nasm -o obj.o -f elf -g -l list.lst hello.asm компилирую исходный файл hello.asm в obj.o, проверяю, что файл создан.(рис. ??)



Компиляция и её проверка

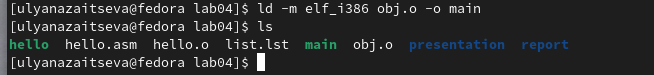
1. Компоновщик LD

Передаю объектный файл на обработку компоновщику с помощью команды ld -m elf\_i386 hello.o -o hello и проверяю, что исполняемый файл hello был создан.(рис. ??)



Компиляция и её проверка

Создаю ещё один файл с помощью команды ld -m elf\_i386 obj.o -o main.(рис. ??)



Создание файла

1. Запуск исполняемого файла

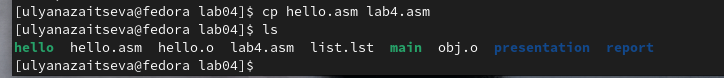
Запускаю созданный исполняемый файл с помощью команды ./hello.(рис. ??)

Запуск файла

Запуск файла

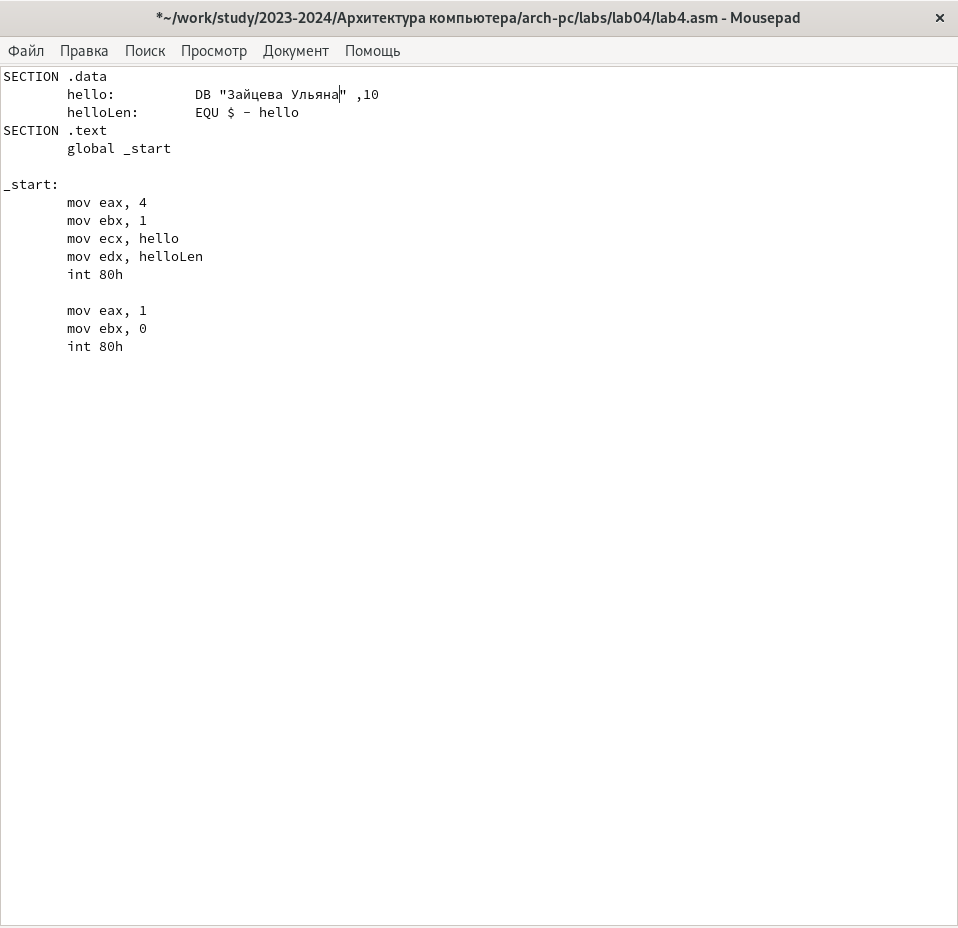
1. Задание для самостоятельной работы

В каталоге ~/work/arch-pc/‘Архитектура компьютера’/arch-pc/labs/lab04 создаю копию файла hello.asm с именем lab4.asm с помощью команды cp(рис. ??)



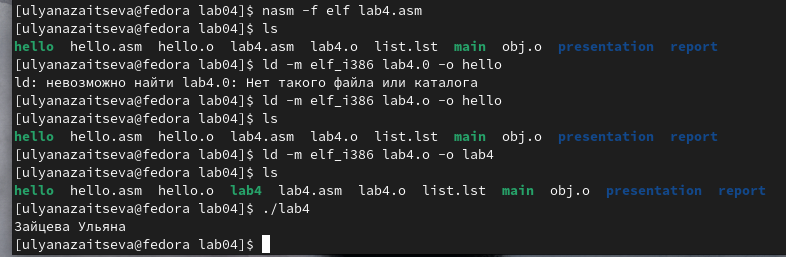
Копирование файла

С помощью 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.(рис. ??)



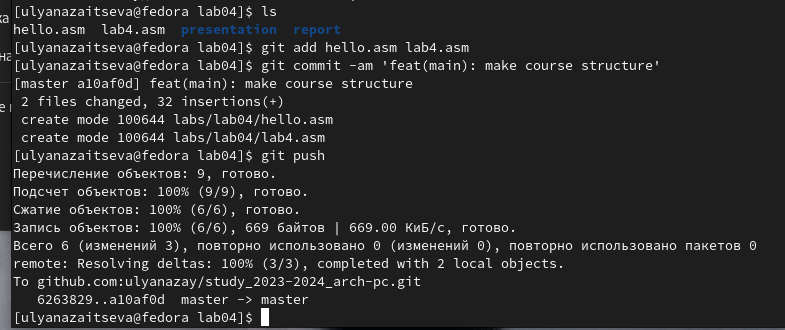
Вношу изменения в текст программы

Компилирую полученный текст программы lab4.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл.(рис. ??)



Вношу изменения в текст программы

Проверяю что файлы находятся в нужном каталоге и загружаю на Гитхаб(рис. ??)



Загрузка файлов на Github

# 5 Выводы

В ходе выполнения лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM