Отчет по лабораторной работе №4

Дисциплина: Архитектура компьютера

Карпова Анастасия Александровна

Содержание

[1 Цель работы 1](#_Toc148529199)

[2 Задание 1](#_Toc148529200)

[3 Теоретическое введение 1](#_Toc148529201)

[4 Выполнение лабораторной работы 2](#_Toc148529202)

[5 Выводы 5](#_Toc148529203)

[Список литературы 5](#_Toc148529204)

# 1 Цель работы

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

# 2 Задание

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

**Создание программы Hello world!**

Создаю каталог для работы с программами на языке ассемблера NASM и перехожу в созданный каталог (рис. 1).

Создание каталога и перемещение

Рис. 1. Создание каталога и перемещение

Создаю текстовый файл с именем hello.asm и открываю его в gedit (рис. 2)

Создание текстового файла

Рис. 2. Создание текстового файла

Ввожу следующий текст (рис. 3)

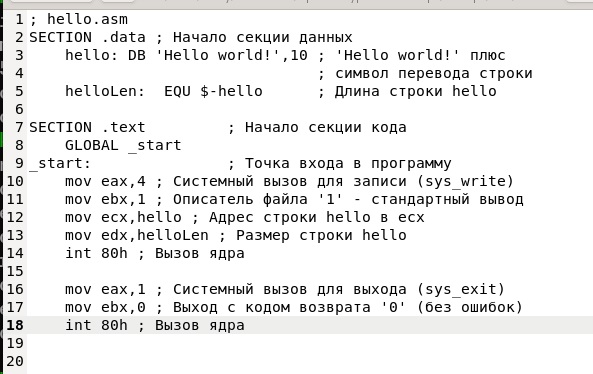


Рис. 3. Ввод текста

**Работа с транслятором NASM**

NASM превращает текст программы в объектный код. Для компиляции приведённого выше текста программы «Hello World» необходимо написать следующее. (рис. 4)

Компиляция

Рис. 4. Компиляция

Расширенный синтаксис командной строки NASM

Ввожу команду, которая скомпилирует файл hello.asm в файл obj.o (рис. 5)

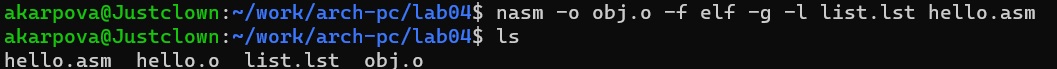


Рис. 5. Компиляция hello.asm в файл obj.o

**Раота с компоновщиком LD**

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить файд hello (рис. 6)

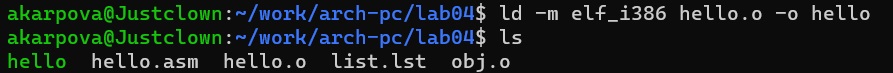


Рис. 6. Передача файла

Ключ -o с последующим значением задаёт в данном случае имя создаваемого исполняемого файла. Поэтому ввожу следующую команду (рис. 7)

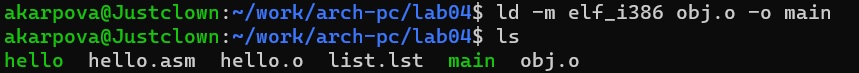


Рис. 7. Передача файла

Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл, находящийся в текущем каталоге набрав в командной строке следующее (рис. 8)

Запуск исполняемого файла

Рис. 8. Запуск исполняемого файла

**Выполнение заданий для самостоятельной работы**

В каталоге ~/work/arch-pc/lab04 с помощью команды cp создайте копию файла

hello.asm с именем lab4.asm и открываю lab4.asm при помощи gedit, редактирую (рис. 9) (рис. 10)

Копирование

Рис. 9. Копирование

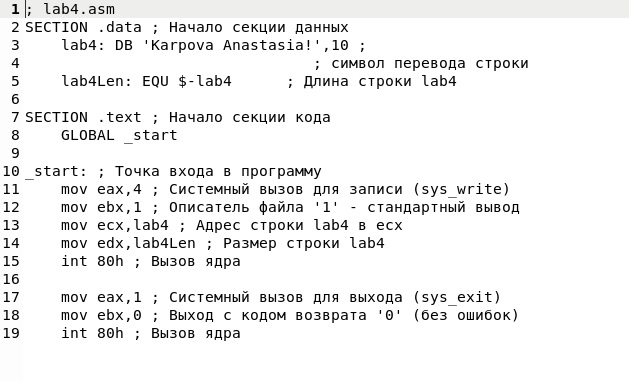


Рис. 10. Запуск и редактирование

Компилирую текст программы в объектный файл (рис. 11)

Компиляция

Рис. 11. Компиляция

Передаю объектный файл lab4.o на обработку компоновщику LD (рис. 12)

Передача файла

Рис. 12. Передача файла

Запускаю исполняемый файл. (рис. 13)

Запуск исполняемого файла

Рис. 13. Запуск исполняемого файла

Отправляю все файлы на git (рис. 14)

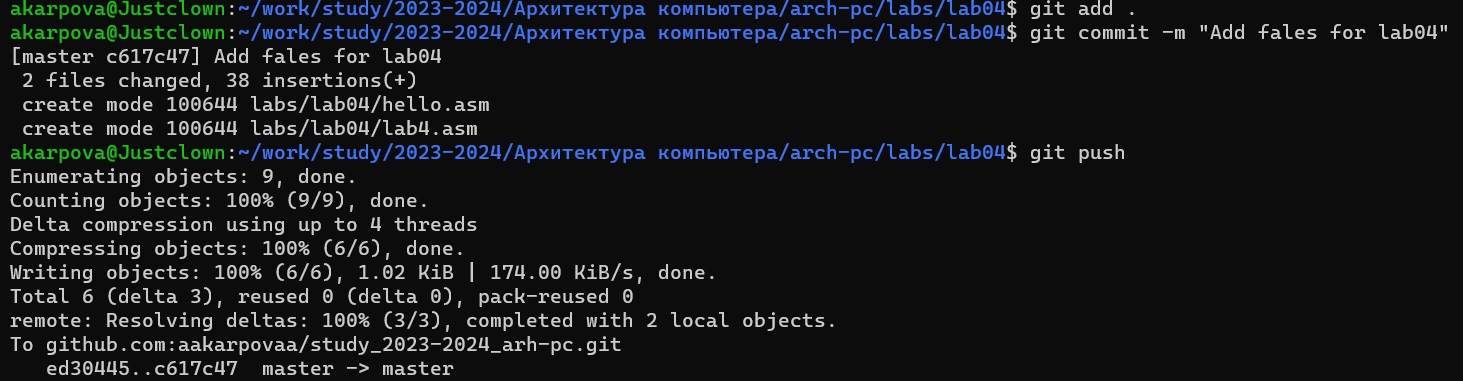


Рис. 14. Отправка файлов на github

# 5 Выводы

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

# Список литературы

1. Архитектура ЭВМ