Отчет по лабораторной работе №4

Дисциплина: Архитектура компьютеров

Малинина Анастасия Игоревна

Содержание

# 1 Цель работы

Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 2 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.

# 3 Теоретическое введение

Основными функциональными элементами любой 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 плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора входят следующие устройства:

1. рифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти;
2. устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера;
3. 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 каче- стве 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

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:

1. устройства внешней памяти, которые предназначены для долговременного хранения больших объёмов данных.
2. 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.

# 4 Выполнение лабораторной работы

## 4.1 Создание программы Hello world!

Создаю каталог для работы с программами на языке ассемблера NASM:(рис. 1).

Рис. 1: создание каталога

Рис. 1: создание каталога

Перехожу в созданный каталог. Создаю текстовый файл с именем hello.asm и открываю этот файл с помощью любого текстового редактора, например, gedit. (рис. 2).

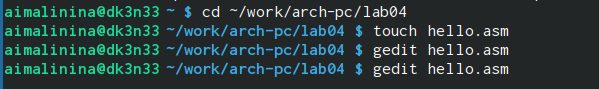


Рис. 2: создание и открытие текстового файла

Ввожу в него следующий текст:(рис. 3).

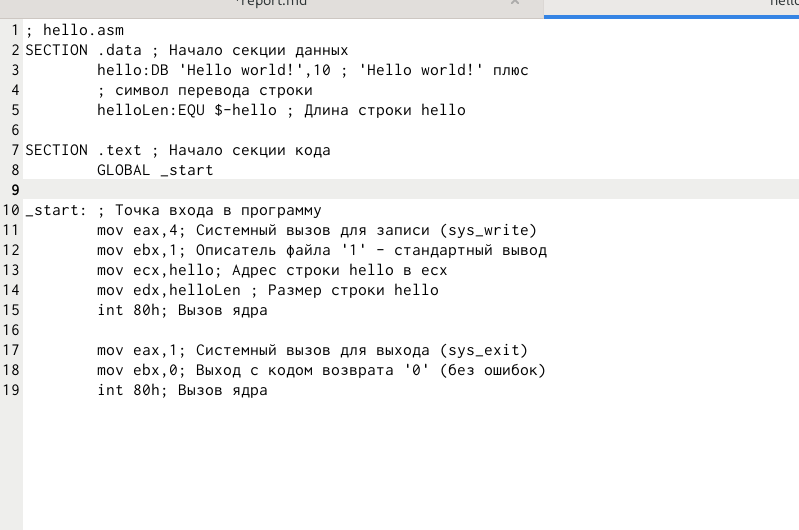


Рис. 3: ввод текста

NASM превращает текст программы в объектный код. Например, для компиляции приве- дённого выше текста программы «Hello World» необходимо написать:(рис. 4).

Рис. 4: ввод команды

Рис. 4: ввод команды

С помощью команды ls проверяю, что объектный файл был создан. (рис. 5).

Рис. 5: проверка

Рис. 5: проверка

Выполняю следующую команду, которая скомпилирует исходный файл hello.asm в obj.o: (рис. 6).

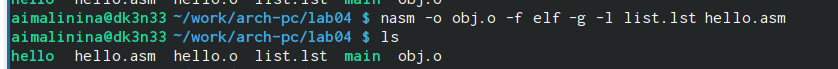


Рис. 6: компиляция файла

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику:(рис. 7).

Рис. 7: обработка

Рис. 7: обработка

Выполняю следующую команду:(рис. 8).

Рис. 8: ввод команды

Рис. 8: ввод команды

## 4.2 Запуск исполняемого файла.

Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке:(рис. 9).

Рис. 9: запуск файла

Рис. 9: запуск файла

## 4.3 Задание для самостоятельной работы

В каталоге ~/work/arch-pc/lab04 с помощью команды cp создаю копию файла hello.asm с именем lab4.asm :(рис. 10).

Рис. 10: ввод команды

Рис. 10: ввод команды

С помощью текстового редактора вношу изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с фио :(рис. 11).

Рис. 11: изменения в программе

Рис. 11: изменения в программе

Оттранслирую полученный текст программы lab4.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл:(рис. 12).

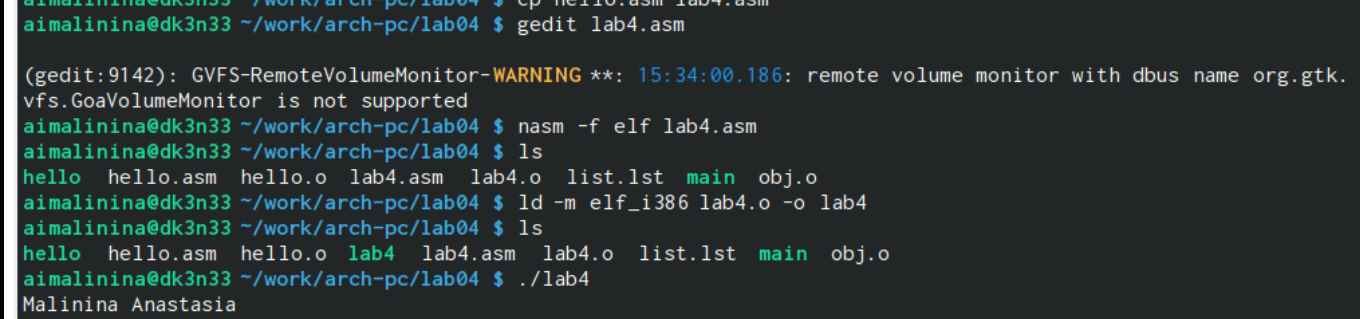


Рис. 12: запуск файла

Копируйю файлы hello.asm и lab4.asm в локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/.:(рис. 13).

Рис. 13: копирование файлов

Рис. 13: копирование файлов

Загружаю файлы на Github.:(рис. 14).

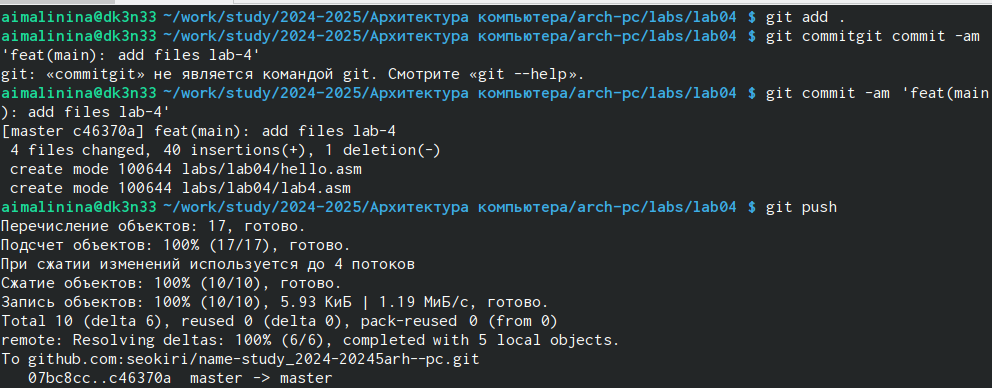


Рис. 14: загрузка

# 5 Выводы

При выполнении данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.