Отчет по лабораорной работе № 4

Дисциплина: архтектура компьютеров

Выслоух Алиса Александровна

Содержание

# 1 Цель работы

Освоение 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 элементами любой ЭВМ являются центральный процессор, память и периферийные устройства. Эти устройства взаимодействуют через общую шину, к которой они подключены. Физическая шина состоит из множества проводников, соединяющих устройства. В современных компьютерах проводники реализованы как электропроводящие дорожки на материнской плате. Главной задачей процессора является обработка данных и координация работы всех узлов компьютера. Центральный процессор включает следующие компоненты:

* арифметико-логическое устройство (АЛУ) — выполняет арифметические и логические операции, необходимые для обработки информации в памяти;
* устройство управления (УУ) — отвечает за управление и контроль всех компьютерных устройств;
* регистры — сверхбыстрая оперативная память небольшого объема для временного хранения промежуточных результатов выполнения команд; регистры делятся на общего назначения и специальные. Знание о регистрах процессора необходимо для программирования на ассемблере. Большинство команд ассемблерных программ используют регистры как операнды. Все команды часто преобразуют данные, хранящиеся в регистрах, что включает пересылку данных между регистрами и памятью или выполнение арифметических и логических операций. Доступ к регистрам осуществляется по именам, а не по адресам, как к основной памяти. Каждый регистр архитектуры x86 имеет название из 2 или 3 латинских букв. Примеры основных регистров общего назначения:
* 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 компонентом ЭВМ является оперативное запоминающее устройство (ОЗУ). Оно — энергозависимое, быстродействующее запоминающее устройство, обеспечивающее взаимодействие с процессором для хранения текущих программ и данных. ОЗУ состоит из пронумерованных ячеек памяти, адреса которых соответствуют хранимым в них данным. Периферийные устройства ЭВМ включают:

* устройства внешней памяти для долговременного хранения больших объемов данных;
* устройства ввода-вывода для обеспечения взаимодействия ЦП с внешней средой.

Вычислительный процесс ЭВМ основан на принципе программного управления, что предполагает, что задачи решаются как последовательность действий, записанных в программе.

Коды команд представляют собой двоичные комбинации из 0 и 1. У каждого машинного кода есть операционная и адресная части. Операционная часть хранит код команды, а адресная — данные или адреса, необходимые для ее выполнения. При каждом выполнении команды процессор выполняет стандартный набор действий, что называется командным циклом. Этот цикл состоит из следующих этапов:

* формирование адреса следующей команды в памяти;
* считывание кода команды из памяти и ее дешифрация;
* выполнение команды;
* переход к следующей команде.

Язык ассемблера (assembly language, сокращенно asm) — это низкоуровневый, машинно-ориентированный язык. NASM — открытый проект ассемблера, версии которого доступны для различных ОС и который позволяет создавать объектные файлы для этих систем. NASM использует Intel-синтаксис и поддерживает 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

С помощью утилиты cd перемещаюсь в каталог, в котором буду работать (рис. 2).

Рис. 2: Перемещение между дерикториями.

Рис. 2: Перемещение между дерикториями.

Создаю в текущем каталоге пустой текстовый файл hello.asm с помощью команды touch (рис. 3)

Рис. 3: Создание файла.

Рис. 3: Создание файла.

Открываю созданный файл в текстовом редакторе gedit и аполняю файл, вставляя в него программу для вывода “Hello word!”(рис. 4).

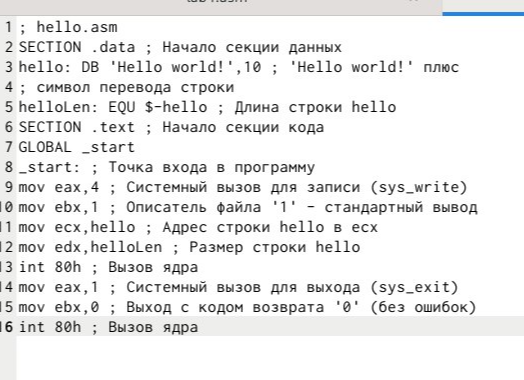


Рис. 4: Открытие файла и вставка программы.

## 4.2 Работа с транслятором NASM

Я преобразую текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду nasm -f elf hello.asm. Параметр -f сообщает транслятору nasm, что необходимо создать бинарный файл в формате ELF (рис. 5). Затем я проверяю успешность выполнения команды с помощью утилиты ls и вижу, что файл “hello.o” действительно создан.(рис. 6)

Рис. 5: Компиляция файла.

Рис. 5: Компиляция файла.

Рис. 6: Проверка

Рис. 6: Проверка

## 4.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует файл hello.asm в файл obj.o, при этом в файл будут включены символы для отладки (ключ -g), также с помощью ключа -l будет создан файл листинга list.lst (рис. 7). Далее проверяю с помощью утилиты ls правильность выполнения команды.

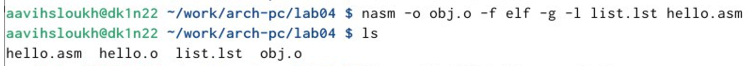


Рис. 7: Компиляция

## 4.4 Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello (рис. 8). Ключ -о задает имя создаваемого исполняемого файла. Далее проверяю с помощью утилиты ls правильность выполнения команды.

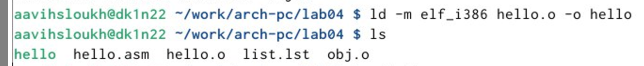


Рис. 8: Передача на обработку

## 4.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello (рис. 9).

Рис. 9: Запуск программы

Рис. 9: Запуск программы

# 5 Задания для самостояельной работы

С помощью утилиты cp создаю в текущем каталоге копию файла hello.asm с именем lab4.asm (рис. 10).

Рис. 10: Создание копии

Рис. 10: Создание копии

С помощью текстового редактора gedit открываю файл lab4.asm и вношу изменения в программу так, чтобы она выводила мои имя и фамилию (рис. 11).

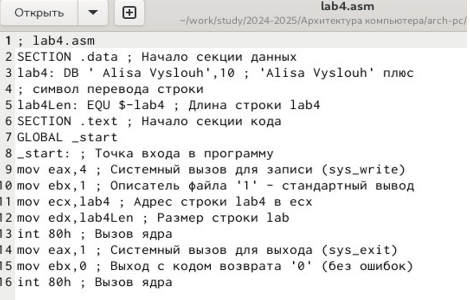


Рис. 11: Открытие файла и его изменение

Компилирую текст программы в объектный файл. Проверяю с помощью утилиты ls, что файл lab4.o создан (рис. 12).

Рис. 12: Компиляция

Рис. 12: Компиляция

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4 (рис. 13).

Рис. 13: Передача файла

Рис. 13: Передача файла

Запускаю исполняемый файл lab5, на экран действительно выводятся мои имя и фамилия (рис. 14).



Рис. 14: Запуск программы

Копирую папку и соединяю ее с lab04. Удаляю лишние файлы после копирования и проверяю с помощью утилиты ls.(рис. 15).

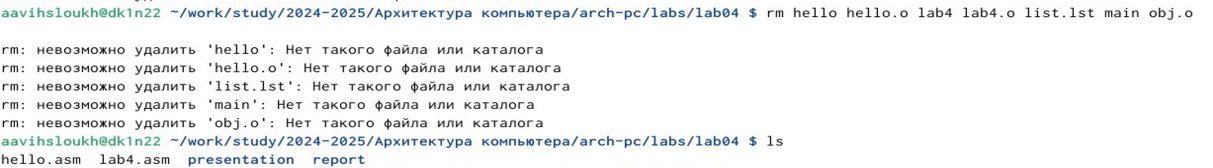


Рис. 15: Удаление файлов и проверка

Ввожу последовательность команд git add . git commit и git push.(рис. 16).

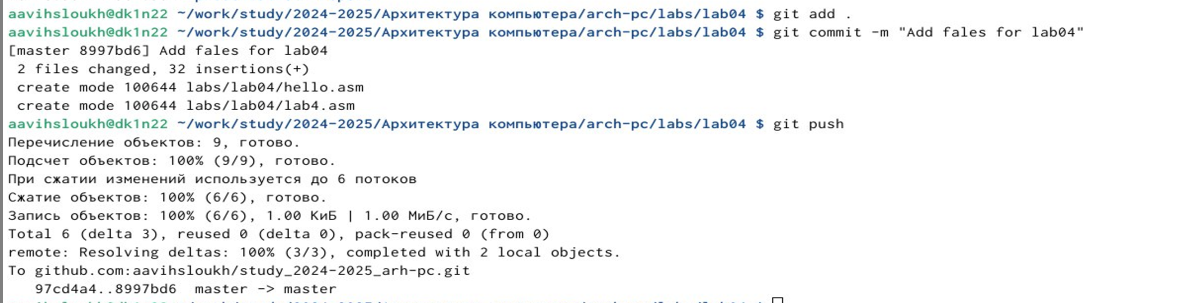


Рис. 16: Загрузка файлов на гитхаб.

# 6 Выводы

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