Шаблон отчёта по лабораторной работе №5

Дисциплина: архитектура компьютера

Пронякова Ольга Максимовна

Содержание

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассем- блере 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 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 про- водников, 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 орга- низация координации всех узлов компьютера. В состав центрального процес- сора (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, вхо- дящая 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 (арифме- тические 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 назначе- ния (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 реги- стров). Например, AH (high AX) — старшие 8 бит регистра AX, AL (low AX) — младшие 8 бит регистра AX.

# 4 Выполнение лабораторной работы

## 4.1 Создание программы Hello world!

С помощью cd перемещаюсь в каталог, в котором буду работать. Создаю в текущем каталоге текстовый файл hello.asm с помощью touch (рис. 1).

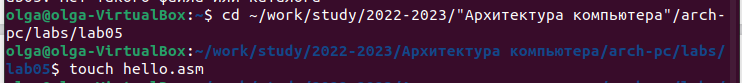


Рис. 1: Создание пустого файла

Открываю созданный файл в текстовом редакторе gedit (рис. 2).

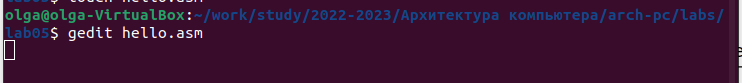


Рис. 2: Открытие файла в текстовом редакторе

Заполняю файл, вставляя в него программу для вывода “Hello world!” (рис. 3).



Рис. 3: Заполнение файла

## 4.2 Работа с транслятором NASM

Превращаю текст программы в объектный код. Для компиляции текста программы «Hello World» использую команду nasm -f elf hello.asm. Далее проверяю правильность выполнения команды с помощью ls (рис. 4).

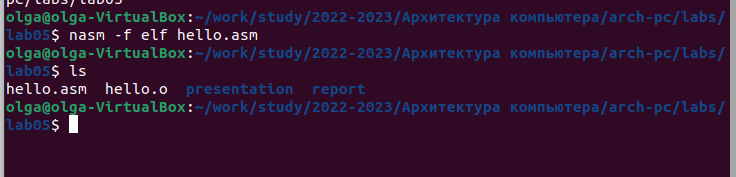


Рис. 4: Комптляция текста программы

## 4.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует файл hello.asm в файл obj.o, такжу с помощью -l будет создан файл list.lst. Проверяю правильность выполнения команды с помощью ls (рис. 5).

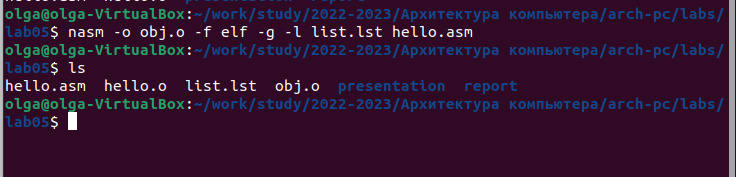


Рис. 5: Комптляция текста программы

## 4.4 Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello. Проверяю правильность выполнения команды с помощью ls (рис. 6).

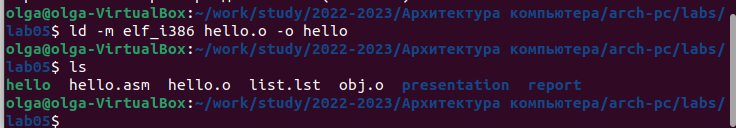


Рис. 6: Передачв объектного файла на обработку компоновщику

Выполняю следующую команду. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o. Проверяю правильность выполнения команды с помощью ls (рис. 7).

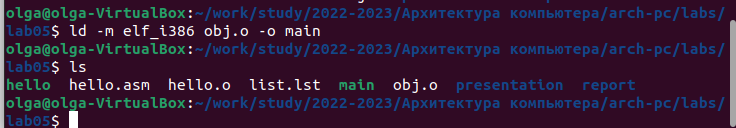


Рис. 7: Передача объектного файла на обработку компоновщику

## 4.5 Запуск исполняемого файла

Запускаю на выполнение исполняемый файл hello (рис. 8).

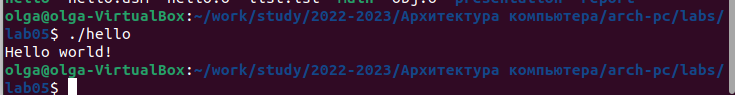


Рис. 8: Запуск исполняемого файла

# 5 Задание для самостоятельной работы

Создаю с помощью cp копию файла hello.asm с именем lab5.asm и открываю его в текстовом редакторе gedit (рис. 9).

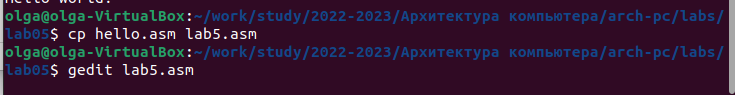


Рис. 9: Создание копии файла

Вношу в программу изменения, чтобы она выводила мои имя и фамилию. Далее компилирую текст программы в объектный файл. Проверяю правильность выполнения команды с помощью ls (рис. 10).

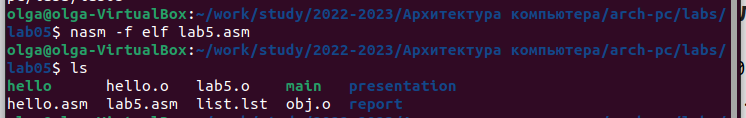


Рис. 10: Компиляция текста программы

Передаю объектный файл lab5.o на обработку компоновщику LD, чтобы получить исполняемый файл lab5 (рис. 11).

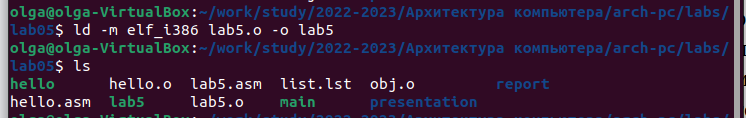


Рис. 11: Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл (рис. 12).

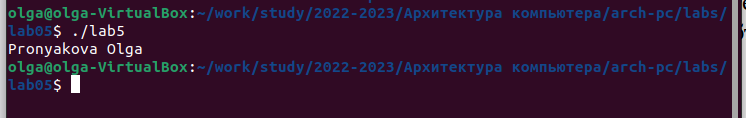


Рис. 12: Запуск исполняемого файла

Добавляем файлы на Github.

# 6 Выводы

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

# Список литературы

[1. Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/1584622/mod_resource/content/1/Лабораторная%20работа%20№3.pdf)