Отчёт по лабораторной работе 4

Архитектура компьютеров и операционные системы

Горелашвили Лия Михайловна НКАбд-02-23

Содержание

# 1 Цель работы

Целью работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# 2 Задания

1. Изучить основы языка Ассемблера
2. Изучить и рассмотреть на практике процесс сборки программы
3. Выполнить задание по программе
4. Подготовить отчет и загрузить на GitHub

# 3 Теоретическое введение

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. Заметим, что получить полный доступ к ресурсам компьютера в современных архитектурах нельзя, самым низким уровнем работы прикладной программы является обращение напрямую к ядру операционной системы. Именно на этом уровне и работают программы, написанные на ассемблере. Но в отличие от языков высокого уровня ассемблерная программа содержит только тот код, который ввёл программист. Таким образом язык ассемблера — это язык, с помощью которого понятным для человека образом пишутся команды для процессора.

В процессе создания ассемблерной программы можно выделить четыре шага:

* Набор текста программы в текстовом редакторе и сохранение её в отдельном файле. Каждый файл имеет свой тип (или расширение), который определяет назначение файла. Файлы с исходным текстом программ на языке ассемблера имеют тип asm.
* Трансляция — преобразование с помощью транслятора, например nasm, текста программы в машинный код, называемый объектным. На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором. Тип объектного файла — o, файла листинга — lst.
* Компоновка или линковка — этап обработки объектного кода компоновщиком (ld), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно не имеет расширения. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение map.
* Запуск программы.

# 4 Выполнение лабораторной работы

## 4.1 Программа Hello world!

Создала каталог с названием lab04 используя команду mkdir. Затем перешла в созданный каталог, воспользовавшись командой cd. Внутри каталога создала файл с именем hello.asm, где написала программу.

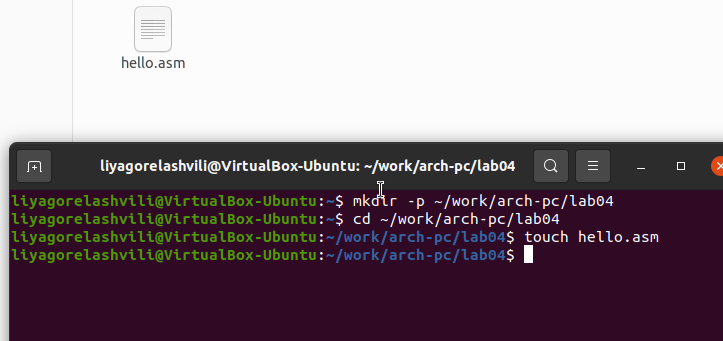


Figure 1: Создан каталог для работы и файл для программы

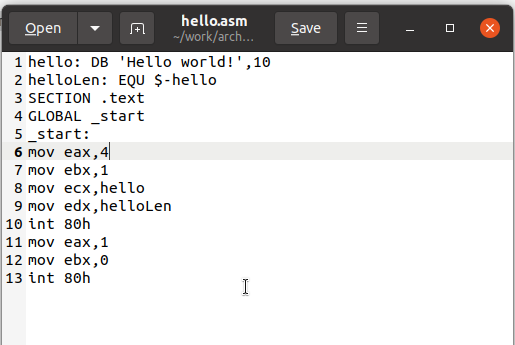


Figure 2: Редактирование файла hello.asm

## 4.2 Транслятор NASM

Использовала транслятор NASM, чтобы преобразовать текст программы в объектный код. Если текст программы был написан без ошибок, то транслятор преобразовал текст программы из файла hello.asm в объектный код и сохраняет его в файле hello.o.

Транслировала файл с помощью команды nasm и получила объектный файл hello.o.

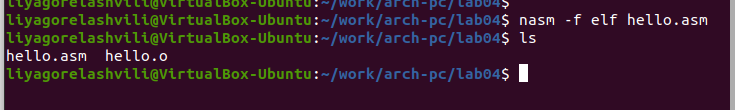


Figure 3: Трансляция программы

## 4.3 Расширенный синтаксис командной строки NASM

Полный вариант командной строки для трансляции с использованием NASM выглядит следующим образом

nasm [-@ косвенный\_файл\_настроек] [-o объектный\_файл] [-f формат\_объектного\_файла] [-l листинг] [параметры...] [--] исходный\_файл

Транслировала файл с помощью команды nasm и использовала дополнительные опции. С опцией -l получила файл листинга с именем list.lst, с опцией -f получила объектный файл с именем obj.o, а с опцией -g в программу добавилась отладочная информация.

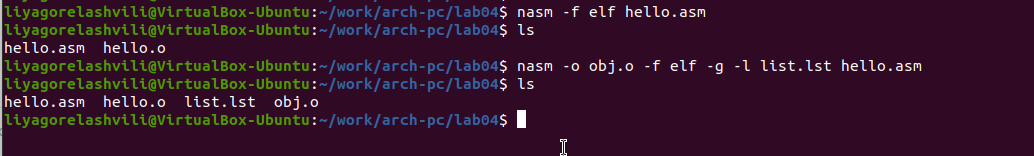


Figure 4: Трансляция программы с дополнительными опциями

## 4.4 Компоновщик LD

Для получения исполняемой программы, объектный файл нужно передать на обработку компоновщику.

Выполнила команду ld и получила исполняемый файл с именем hello из объектного файла hello.o.

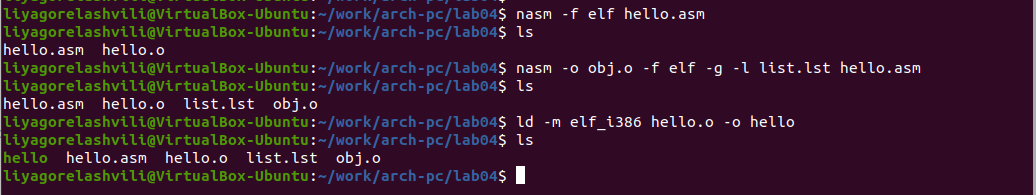


Figure 5: Компоновка программы

Еще раз выполнена команда ld для объектного файла obj.o и получен исполняемый файл с именем main

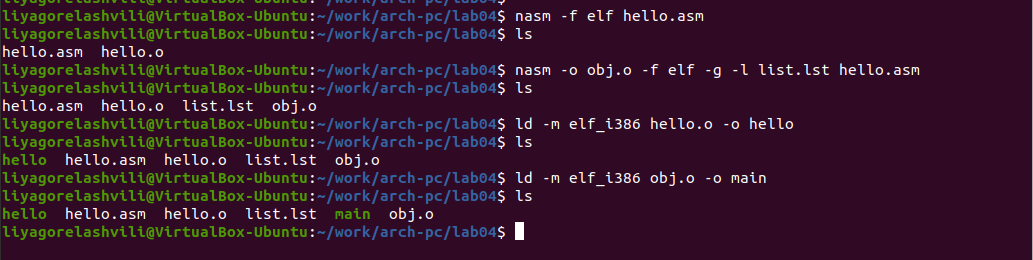


Figure 6: Компоновка программы

## 4.5 Запуск исполняемого файла

Запустила исполняемые файлы.



Figure 7: Запуск программы

## 4.6 Задание для самостоятельной работы

Скопировала файл hello.asm в файл lab4.asm.

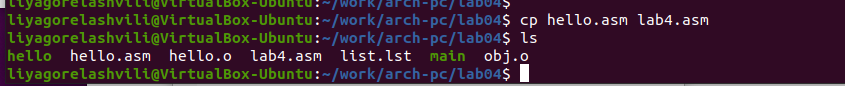


Figure 8: Скопировала файл

Изменила сообщение Hello world на свое имя.

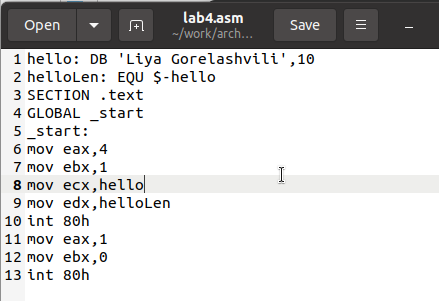


Figure 9: Редактирование файла lab4.asm

Запустила программу и проверила.

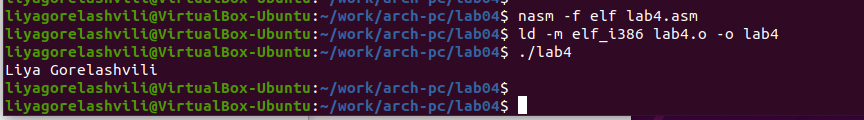


Figure 10: Тестирование программы lab4.asm

# 5 Выводы

Освоил процесс компиляции и сборки программ, написанных на ассемблере nasm.