Отчёт по лабораторной работе №5

Создание и процесс обработки программ на языке ассемблера NASM

Аскеров Алекcандр Эдуардович

Содержание

[1 Цель работы 1](#__RefHeading___Toc278_515706175)

[2 Выполнение лабораторной работы 1](#__RefHeading___Toc280_515706175)

[2.1 Программа Hello world! 1](#__RefHeading___Toc282_515706175)

[2.2 Транслятор NASM 3](#__RefHeading___Toc284_515706175)

[2.3 Расширенный синтаксис командной строки NASM 3](#__RefHeading___Toc286_515706175)

[2.4 Компоновщик LD 4](#__RefHeading___Toc288_515706175)

[2.5 Запуск исполняемого файла 4](#__RefHeading___Toc290_515706175)

[2.6 Задание для самостоятельной работы 4](#__RefHeading___Toc292_515706175)

[3 Выводы 6](#__RefHeading___Toc294_515706175)

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

## 2.1 Программа Hello world!

Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран.

Создадим каталог для работы с программами на языке ассемблера NASM.

Рис. 1: Создание каталога для работы с программами

Рис. 1: Создание каталога для работы с программами

Перейдём в созданный каталог.

Рис. 2: Переход в каталог lab05

Рис. 2: Переход в каталог lab05

Создадим текстовый файл с именем hello.asm.

Рис. 3: Создание файла hello.asm

Рис. 3: Создание файла hello.asm

Откроем этот файл с помощью текстового редактора gedit и введём в него следующий текст.

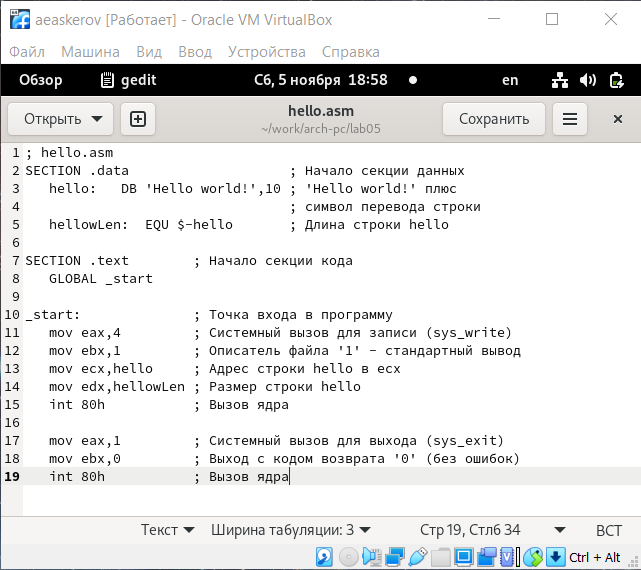


Рис. 4: Ввод текста в файл hello.asm

## 2.2 Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать: nasm -f elf hello.asm.

Если текст программы набран без ошибок, то транслятор преобразует текст программы из файла hello.asm в объектный код, который запишется в файл hello.o. Таким образом, имена всех файлов получаются из имени входного файла и расширения по умолчанию. При наличии ошибок объектный файл не создаётся, а после запуска транслятора появятся сообщения об ошибках или предупреждения.

С помощью команды ls проверим, что объектный файл был создан. Получившийся объектный файл имеет имя hello.

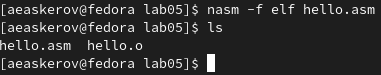


Рис. 5: Компиляция hello.asm и создание объектного файла

NASM не запускают без параметров. Ключ -f указывает транслятору, что требуется создать бинарные файлы в формате ELF. Следует отметить, что формат elf64 позволяет создавать исполняемый код, работающий под 64-битными версиями Linux. Для 32-битных версий ОС указываем в качестве формата просто elf.

NASM всегда создаёт выходные файлы в текущем каталоге.

## 2.3 Расширенный синтаксис командной строки NASM

Скомпилируем исходный файл hello.asm в obj.o (опция - o позволяет задать имя объектного файла, в данном случае obj.o), при этом формат выходного файла будет elf, и в него будут включены символы для отладки (опция -g), кроме того, будет создан файл листинга list.lst (опция -l). Для этого выполним следующую команду.

Рис. 6: Компиляция исходного файла hello.asm в obj.o, а также создание файла листинга list.lst

Рис. 6: Компиляция исходного файла hello.asm в obj.o, а также создание файла листинга list.lst

С помощью команды ls проверим, что файлы были созданы.

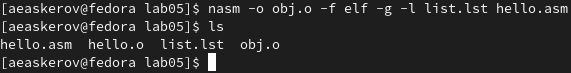


Рис. 7: Проверка того, что файлы созданы

## 2.4 Компоновщик LD

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику. С помощью команды ls проверим, что исполняемый файл hello был создан.

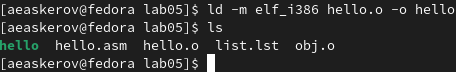


Рис. 8: Передача объектного файла на обработку компоновщику

Выполним следующую команду: ld -m elf\_i386 obj.o -o main.

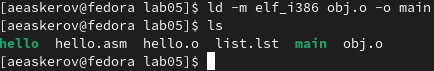


Рис. 9: Передача объектного файла на обработку компоновщику

Исполняемый файл будет иметь имя main, а объектный файл, из которого собран этот исполняемый файл, будет иметь имя obj.o.

## 2.5 Запуск исполняемого файла

Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке: ./hello.

Рис. 10: Запуск исполняемого файла на исполнение

Рис. 10: Запуск исполняемого файла на исполнение

## 2.6 Задание для самостоятельной работы

1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создадим копию файла hello.asm с именем lab5.asm.

Рис. 11: Переименование файла hello.asm в lab5.asm

Рис. 11: Переименование файла hello.asm в lab5.asm

1. С помощью любого текстового редактора внесём изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран выводилась строка с фамилией и именем.

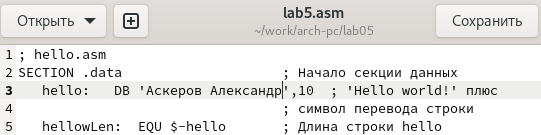


Рис. 12: Внесение изменений в файл lab5.asm

1. Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

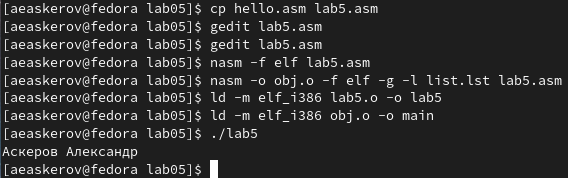


Рис. 13: Трассировка, компоновка и запуск файла lab5.asm

1. Скопируем файлы hello.asm и lab5.asm в локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/archpc/labs/lab05/. Загрузим файлы на Github.

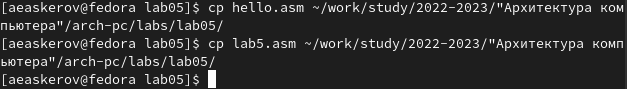


Рис. 14: Копирование файлов hello.asm и lab5.asm

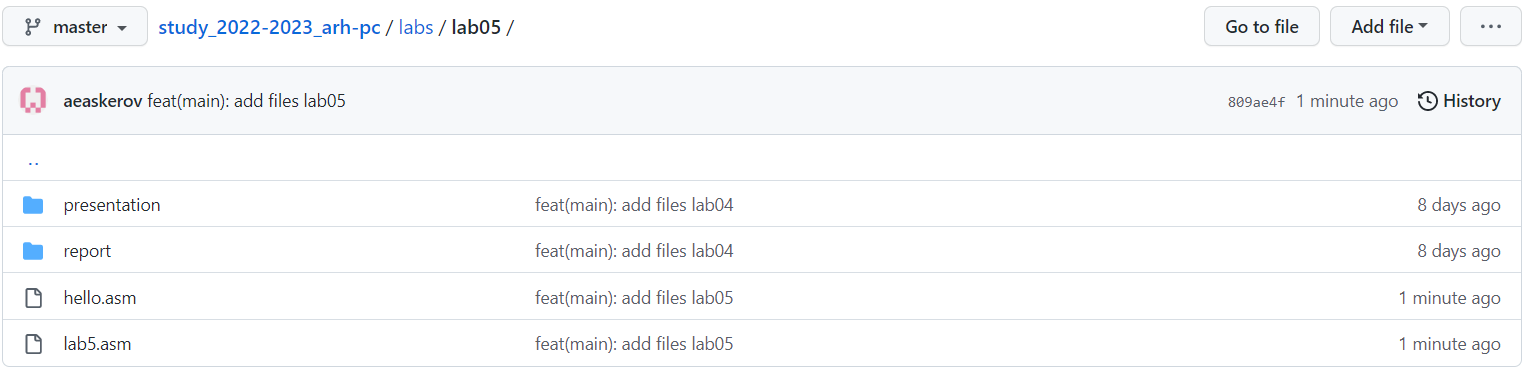


Рис. 15: Файлы, загруженные на GitHub

# 3 Выводы

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