Отчёт по лабораторной работе №5

Уткина Алина Дмитриевна

Содержание

# 1 Цель работы

Целью данной работы является освоение 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.

Для этого переходим в каталог для работы с программами на языке ассемблера NASM, создаем там текстовый файл с именем hello.asm и открываем его с помощью текстового редактора gedit (рис. 1).

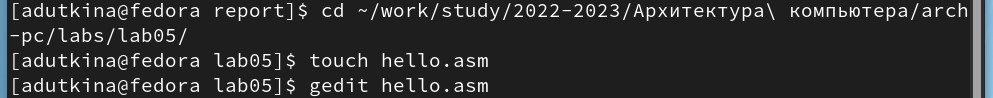


Рис. 1: Создание файла hello.asm

Далее в файл вводим следующий текст (рис. 2):

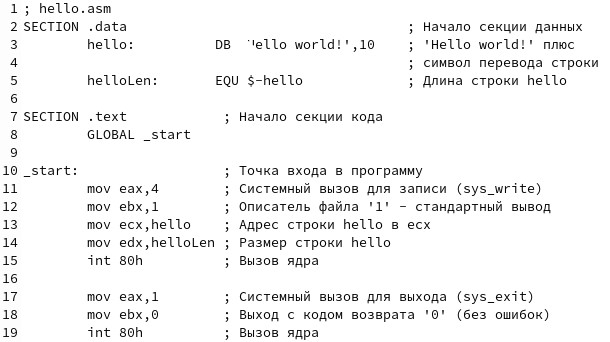


Рис. 2: Текст файла hello.asm

В отличие от многих современных высокоуровневых языков программирования, в ассемблерной программе каждая команда располагается на отдельной строке. Размещение нескольких команд на одной строке недопустимо. Синтаксис ассемблера NASM является чувствительным к регистру, т.е. есть разница между большими и малыми буквами.

## 2.2 Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать определенную команду (рис. 3):



Рис. 3: Компиляция программы

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. 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

Полный вариант командной строки nasm выглядит следующим образом: nasm [-@ косвенный\_файл\_настроек] [-o объектный\_файл] [-f формат\_объектного\_файла] [-l листинг] [параметры…] [–] исходный\_файл

Выполним следующую команду (рис. 4):

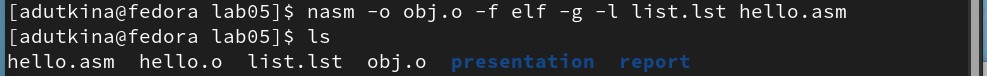


Рис. 4: Компиляция программы с дополнительными опциями

Данная команда скомпилирует 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).

## 2.4 Компоновщик LD

Как видно из схемы (рис. 5), чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику (рис. 6):



Рис. 5: Процесс создания ассемблерной программы

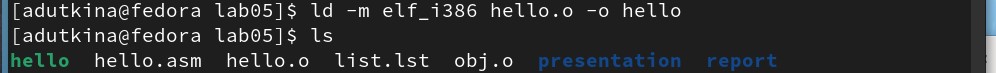


Рис. 6: Передача файла на обработку компоновщику

Компоновщик ld не предполагает по умолчанию расширений для файлов, но принято использовать следующие расширения: \* o – для объектных файлов; \* без расширения – для исполняемых файлов; \* map – для файлов схемы программы; \* lib – для библиотек

Ключ -o с последующим значением задаёт в данном случае имя создаваемого исполняемого файла. Выполним следующую команду (рис. 7)



Рис. 7: Применение команды ld

Здесь исполняемый файл с именем main, а объектный файл, из которого собран исполняемый файл - obj.o

## 2.5 Запуск исполняемого файла

Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке ./hello (рис. 8)

Рис. 8: Запуск исполняемого файла

Рис. 8: Запуск исполняемого файла

## 2.6 Задания для самостоятельной работы

1. В каталоге ~/…/lab05 с помощью команды cp создадим копию файла hello.asm с именем lab5.asm (рис. 9)

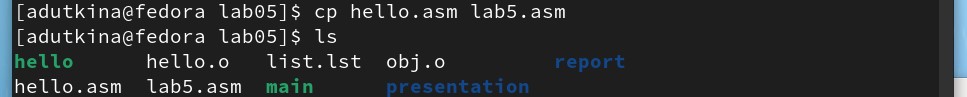


Рис. 9: Копирование файла с новым именем

1. С помощью любого текстового редактора внесем изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран выводилась строка с фамилией и именем (рис. 10).

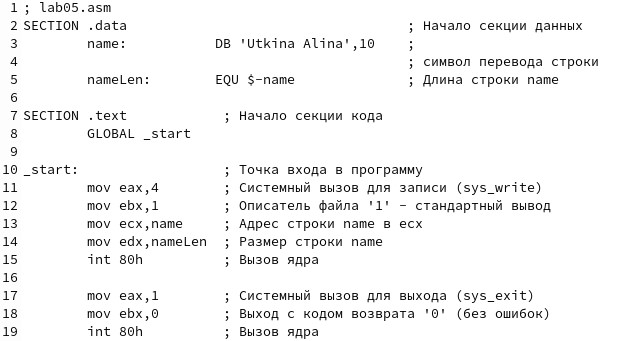


Рис. 10: Внесение изменений в объектный файл

1. Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл (рис. 11).

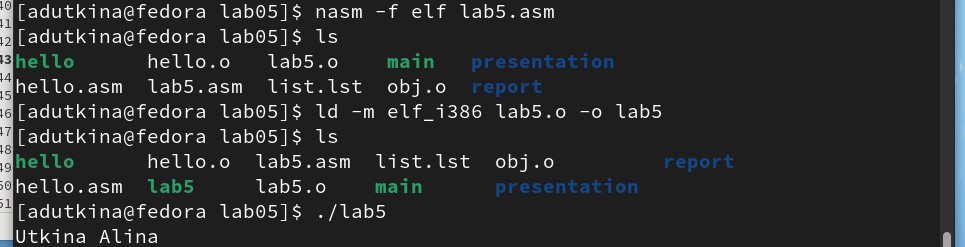


Рис. 11: Трансляция, компоновка и запуск программы

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. 12).

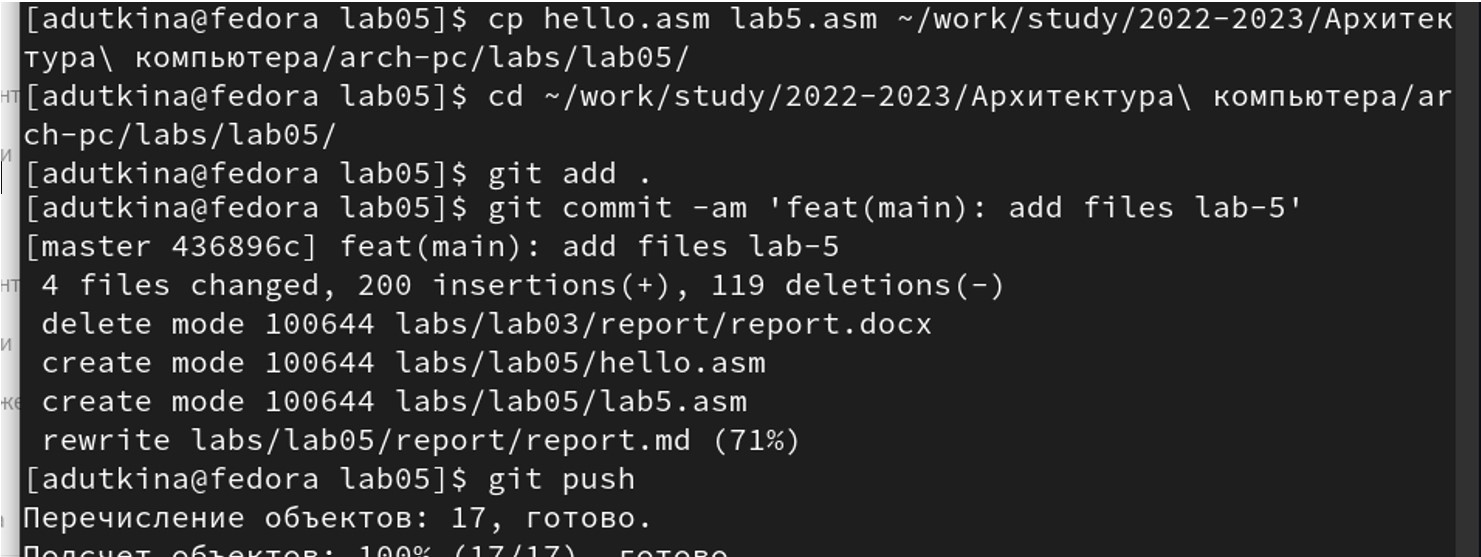


Рис. 12: Копирование файлов и их загрузка на Github

# 3 Выводы

В ходе лабораторной работы были изучены теоретические сведения о языке ассемблера и освоены процедуры компиляции и сборки программ, написанных на ассемблере NASM.