Отчёт по лабораторной работе № 5

Архитектура компьютера

Миразим Азимов

Содержание

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на, машинноориентированный языке низкого уровня, ассемблере NASM.

# 2 Задание

1. Создать файлы расширения .asm.
2. Отредактировать .asm файлы.
3. Оттранслировать .asm файлы в объектные.
4. С помощью компановщика создать исполняемые файлы и запустить.

# 3 Выполнение лабораторной работы

## 3.1 Программа Hello world!

Рассмотрели пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран. Создали каталог для работы с программами на языке ассемблера NASM: (рис. 1)

Рис. 1: Созданный каталог

Рис. 1: Созданный каталог

Перешли в созданный каталог. (рис. 2)

Рис. 2: Переход в каталог

Рис. 2: Переход в каталог

Создали текстовый файл с именем hello.asm. Открыли этот файл с помощью текстового редактора gedit. Введите в него следующий текст: (рис. 3)

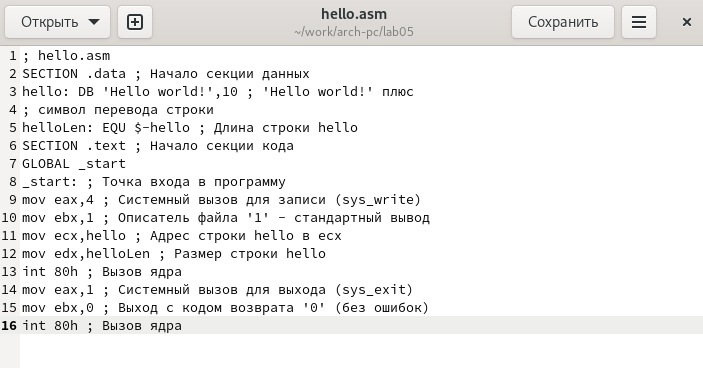


Рис. 3: gedit

## 3.2 Транслятор NASM

NASM превращает текст программы в объектный код. Для компиляции текста программы «Hello World» написали: (рис. 4)

Рис. 4: Компиляция

Рис. 4: Компиляция

С помощью транслятора преобразовали текст программы из файла hello.asm в объектный код, который записали в файл hello.o С помощью команды ls проверили, что объектный файл был создан. Объектный файл имеет имя hello.o (рис. 5)

Рис. 5: Созданный объектный файл

Рис. 5: Созданный объектный файл

## 3.3 Расширенный синтаксис командной строки NASM

Выполнили следующую команду: (рис. 6)

Рис. 6: obj.o

Рис. 6: obj.o

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).

С помощью команды ls проверьте, что файлы были созданы. (рис. 7)

Рис. 7: Созданные файлы

Рис. 7: Созданные файлы

## 3.4 Компоновщик LD

Для получения исполняемой программы, объектный файл передали на обработку компоновщику: (рис. 8)

Рис. 8: Работа компановщика

Рис. 8: Работа компановщика

С помощью команды ls проверьте, что исполняемый файл hello был создан. (рис. 9)

Рис. 9: Созданный файл hello

Рис. 9: Созданный файл hello

Выполните следующую команду: (рис. 10)

Рис. 10: Компановка файла

Рис. 10: Компановка файла

Исполняемый файл будет иметь имя main. Объектный файл, из которого собран этот исполняемый файл, будет иметь имя main.о (рис. 11)

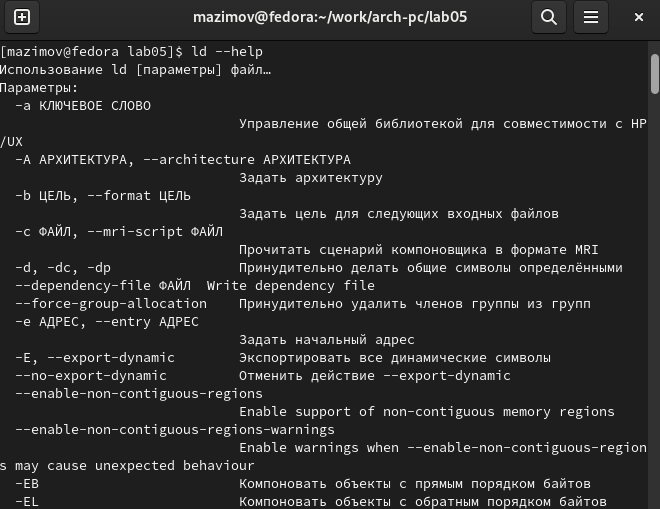


Рис. 11: Проверка названий файлов

Формат командной строки LD увидели, набрав ld –help. (рис. 12)

Рис. 12: ld –help

Рис. 12: ld –help

## 3.5 Запуск исполняемого файла

Запустили на выполнение созданный исполняемый файл, находящийся в текущем каталоге. (рис. 13)

Рис. 13: Выполнение файла

Рис. 13: Выполнение файла

## 3.6 Задание для самостоятельной работы

1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создали копию файла hello.asm с именем lab5.asm (рис. 14), (рис. 15)

Рис. 14: cp lab5.asm

Рис. 14: cp lab5.asm

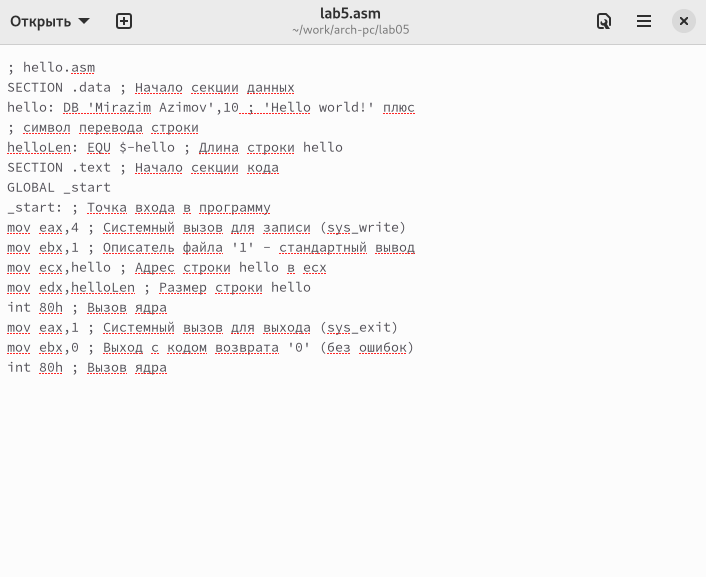


Рис. 15: ls lab05

1. С помощью текстового редактора внесли изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран выводилась строка с фамилией и именем. (рис. 16)

Рис. 16: Изменения в тексте программы

Рис. 16: Изменения в тексте программы

1. Оттранслировали полученный текст программы lab5.asm в объектный файл. (рис. 17)

Рис. 17: lab5.о

Рис. 17: lab5.о

Выполнили компоновку объектного файла и запустили получившийся исполняемый файл. (рис. 18), (рис. 19)

Рис. 18: lab5.о

Рис. 18: lab5.о

Рис. 19: lab5 запуск

Рис. 19: lab5 запуск

1. Скопируйте файлы hello.asm и lab5.asm в Ваш локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/archpc/labs/lab05/. (рис. 20), (рис. 21)

Рис. 20: hello.asm

Рис. 20: hello.asm

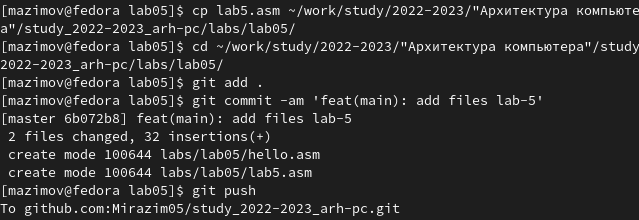


Рис. 21: lab5.asm

Загрузите файлы на Github. (рис. 22)

Рис. 22: Загрузка файлов на Github

# 4 Выводы

В ходе лабораторной работы были освоены процедуры компиляции и сборки программ, написанных на машинноориентированном языке низкого уровня, ассемблере NASM.