Отчёт по лабораторной работе №4

дисциплина: Архитектура вычислительных систем

Мосолов Александр Денисович

Содержание

1 Цель работы **5**

2 Выполнение лабораторной работы **6**
2.1 Программа Hello world! 6
2.2 Транслятор NASM 7
2.3 Расширенный синтаксис командной строки NASM 7
2.4 Компоновщик LD 8
2.5 Запуск исполняемого файла 8
2.6 Задание для самостоятельной работы 8

3 Выводы **11**

Список иллюстраций

2.1 Создаем каталог для работы 6
2.2 Asm-файл 6
2.3 Открываем файл 6
2.4 Вводим текст 7
2.5 Компилируем текст 7
2.6 Присваиваем имя скомпилированному файлу и создаем файл листинга 7
2.7 Передаем файл hello.o на обработку 8
2.8 Передаем файл obj.o на обработку 8
2.9 Запускаем исполняемый файл 8
2.10 Копируем файл hello.asm 8
2.11 Вносим изменения в текст программы 9
2.12 Транслируем текст 9
2.13 Переносим файлы 9
2.14 Загрузка файлов на Github 10

Список таблиц

1 Цель работы

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.

Создаем каталог для работы с программами на языке ассемблера *NASM* (рис. [2.1]):

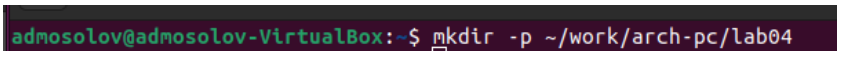

Рис. 2.1: Создаем каталог для работы

Переходим в каталог *lab4* и создаем в нем файл с именем *hello.asm*.

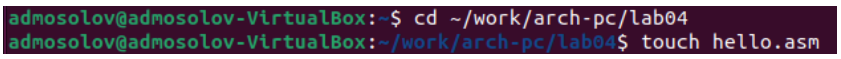

Рис. 2.2: Asm-файл

Открываем этот файл с помощью текстового редактора *gedit*.

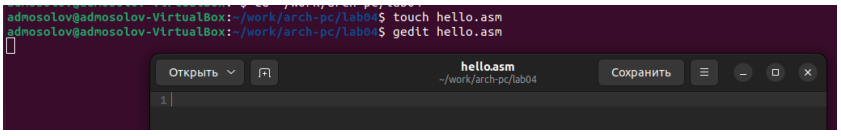

Рис. 2.3: Открываем файл

Вводим в него текст для последующей компиляции.

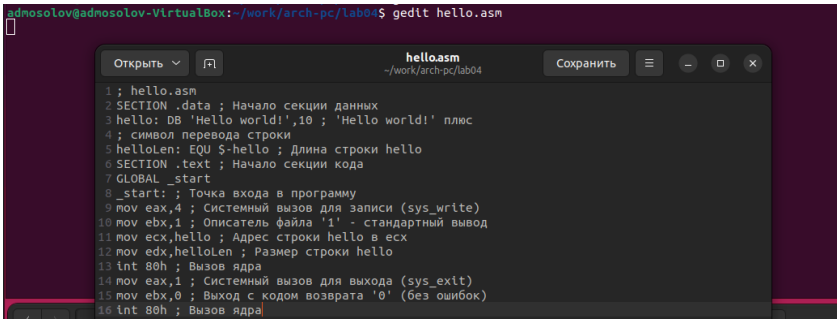

Рис. 2.4: Вводим текст

2.2 Транслятор NASM

NASM превращает текст программы в объектный код. Компилируем приведённый выше текст программы, проверяем наличие нового файла *hello.o*.

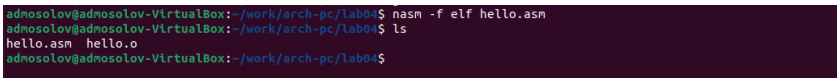

Рис. 2.5: Компилируем текст

2.3 Расширенный синтаксис командной строки NASM

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*).

Проверяем существование созданных файлов с помощью команды *ls*.

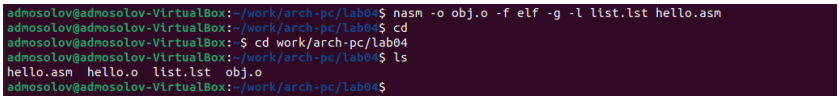

Рис. 2.6: Присваиваем имя скомпилированному файлу и создаем файл листинга

2.4 Компоновщик LD

Объектный файл необходимо передать на обработку компоновщику.

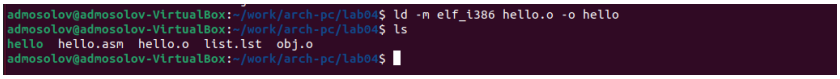

Рис. 2.7: Передаем файл hello.o на обработку

Для тренировки отправляем на обработку компоновщику файл *obj.o* и называем его *main*.

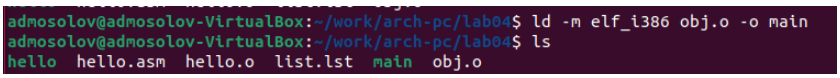

Рис. 2.8: Передаем файл obj.o на обработку

2.5 Запуск исполняемого файла

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

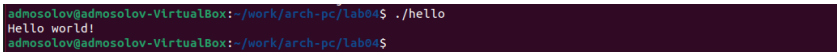

Рис. 2.9: Запускаем исполняемый файл

2.6 Задание для самостоятельной работы

В каталоге *~/work/arch-pc/lab04* с помощью команды *cp* создайте копию файла *hello.asm* с именем *lab4.asm*.

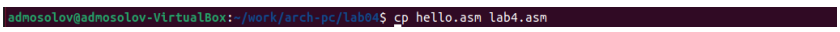

Рис. 2.10: Копируем файл hello.asm

С помощью текстового редактора вносим изменения в текст программы в файле *lab4.asm* так, чтобы вместо *Hello world!* на экран выводилась строка с фамилией и именем.

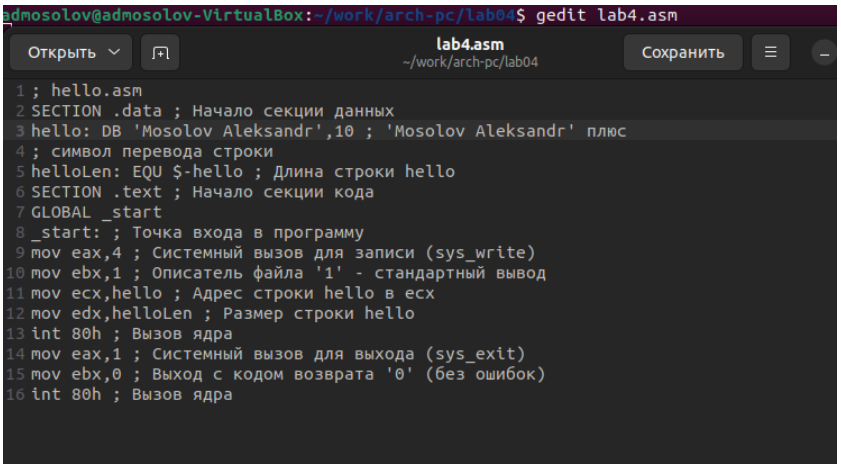

Рис. 2.11: Вносим изменения в текст программы

Транслируем полученный текст программы *lab4.asm* в объектный файл. Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл.

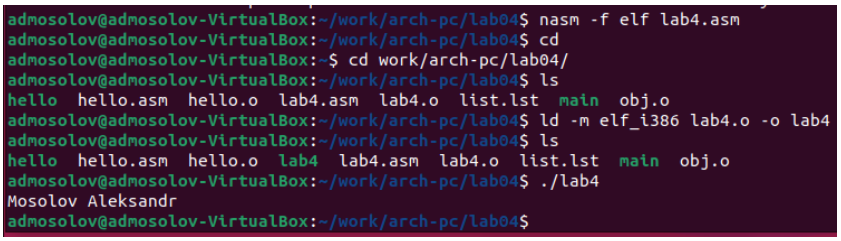

Рис. 2.12: Транслируем текст

Переносим файлы *hello.asm* и *lab4.asm* в локальный репозиторий в ката- лог *~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/*.

Рис. 2.13: Переносим файлы

Загружаем файлы на *Github*.

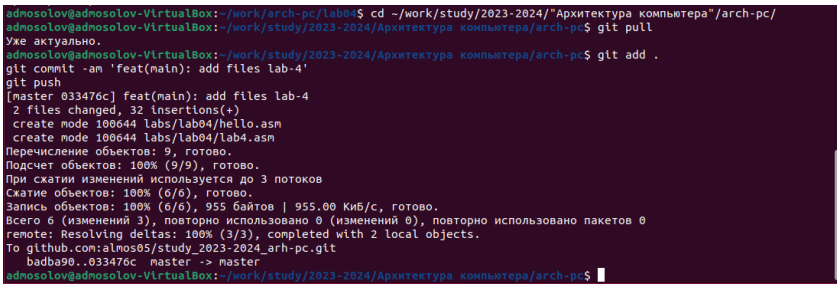

Рис. 2.14: Загрузка файлов на Github

3 Выводы

В ходе выполнения лабораторной работы были освоены процедуры компиляции и сборки программ, написанных на ассемблере *NASM*.