Отчёт по лабораторной работе №4

Дисциплина: Архитектура компьютера

Абакумов Тимофей Александрович

Содержание

1 Цель работы

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

2 Задание

Порядок выполнения лабораторной работы

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла

Задание для самостоятельной работы

1. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создайте копию файла hello.asm с именем lab4.asm
2. С помощью любого текстового редактора внесите изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.

3. Оттранслируйте полученный текст программы lab4.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы hello.asm и lab4.asm в Ваш локальный репозиторий в каталог ~/work/study/2024-2025/“Архитектура компьютера”/arch-pc/labs/lab04/. Загрузите файлы на Github.

3 Выполнение лабораторной работы

3.1 Создание программы Hello world!

С помощью утилиты cd переместимся в каталог, в котором будем работать (рис. 1).

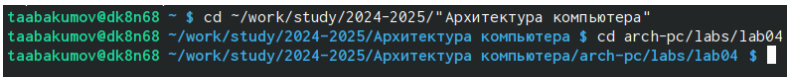

Рис. 1: Перемещение между директориями

Создадим в текущем каталоге пустой текстовый файл hello.asm с помощью утилиты touch (рис. 2).

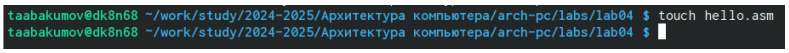

Рис. 2: Создание текстового файла

Откроем созданный файл в текстовом редакторе gedit и вставим туда текст из Лабороторной работы(рис. 3).

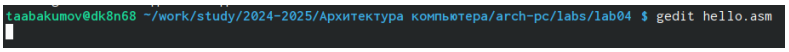

Рис. 3: Работа с текстовым файлом

3.2 Работа с транслятором NASM

Превратим текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду nasm -f elf hello.asm, ключ -f указывает транслятору nasm, что требуется создать бинарный файл в формате ELF (рис. 4).

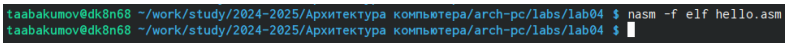

Рис. 4: Работа с транслятором NASM

Далее проверяю правильность выполнения команды с помощью утилиты ls: действительно, создан файл “hello.o”.

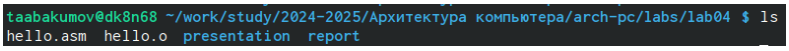

Рис. 5: Проверка правильности

3.3 Работа с расширенным синтаксисом командной строки NASM

Введём команду, которая скомпилирует файл hello.asm в файл obj.o, при этом в файл будут включены символы для отладки (ключ -g), также с помощью ключа -l будет создан файл листинга listМирный.lst (рис. 6). Далее проверим с помощью утилиты ls правильность выполнения команды.

Рис. 6: Работа с расширенным синтаксисом командной строки NASM

3.4 Работа с компоновщиком LD

Передадим объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello (рис. 7). Ключ -o задает имя создаваемого исполняемого файла. Далее проверим с помощью утилиты ls правильность выполнения команды.

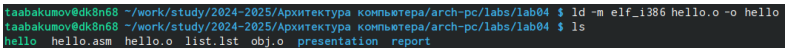

Рис. 7: Работа с компоновщиком LD

Выполним следующую команду (рис. 8). Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o

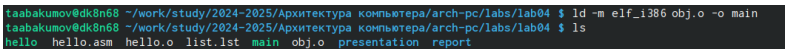

Рис. 8: Передача объектного файла на обработку компоновщику

3.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello (рис. 9).

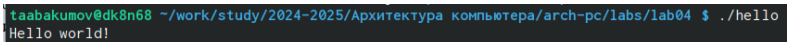

Рис. 9: Запуск исполняемого файла

4 Задание для самостоятельной работы

С помощью утилиты cp создаю в текущем каталоге копию файла hello.asm с именем lab4.asm (рис. 10).

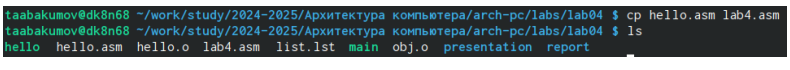

Рис. 10: Создание копии файла

С помощью утилиты mc открываю файл lab4.asm и вношу изменения в программу так, чтобы она выводила мои имя и фамилию. (рис. 11).

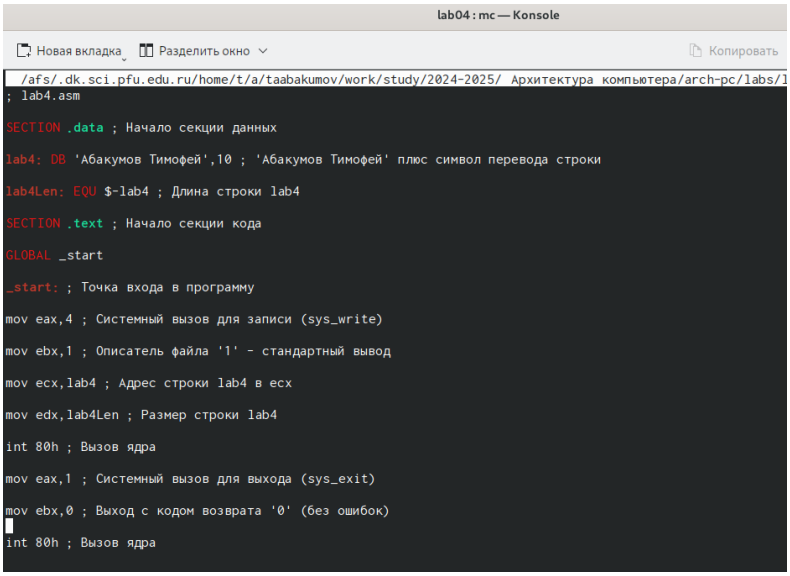

Рис. 11: Изменение програмы

Компилирую текст программы в объектный файл. Затем передадим объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4. Запустим исполняемый файл lab4 (рис. 12).

Рис. 12: Компиляция текста программы

С помощью команд git add . и git commit добавляю файлы на GitHub, комментируя действие как добавление файлов для лабораторной работы №5. После этого отправляем файлы на сервер с помощью команды git push (рис. 13).

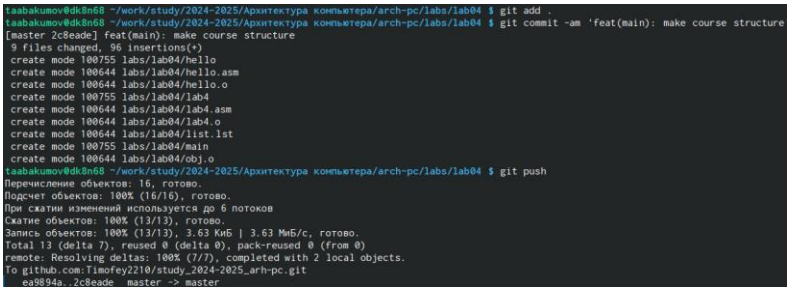

Рис. 13: Отправка файлов

Проверим на GitHub все файлы(рис. 14)

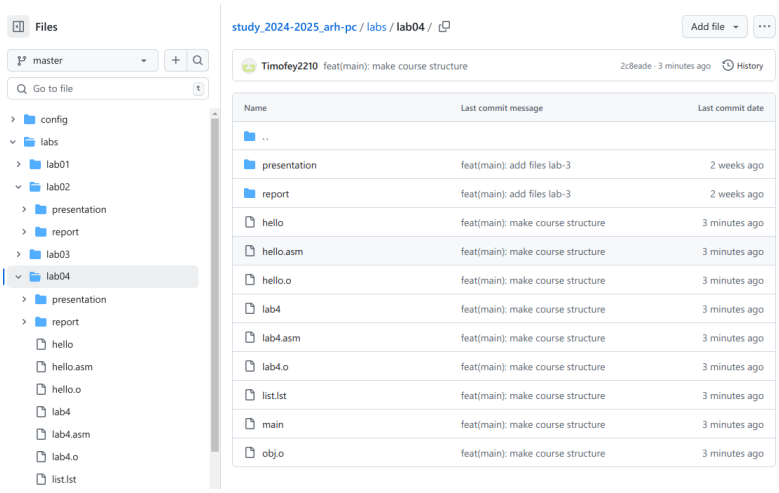

Рис. 14: Проверка файлов

5 Выводы

При выполнении данной лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.