Отчёт по лабораторной работе №4

дисциплина: Архитектура компьютера

Байдина Елизавета Дмитриевна

Содержание

# 1 Цель работы

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

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
6. Выполнение заданий для самостоятельной работы.

# 2 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 приведено краткое описание стандартных каталогов Unix.

Таблица 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно про Unix см. в [1–4].

# 3 Выполнение лабораторной работы

## 3.1 Создание программы Hello world!

С помощью команды cd переходим в каталог,в котором будем работать. (рис. 1).

Рис. 1: Перемещение между директориями

Рис. 1: Перемещение между директориями

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

Рис. 2: Cоздание пустого файла

Рис. 2: Cоздание пустого файла

Открываю созданный файл в текстовом редакторе. (рис. 3).

Рис. 3: Открытие файла в текстовом редакторе

Рис. 3: Открытие файла в текстовом редакторе

Заполняю файл, вставляя в него программу для вывода “Hello word!” (рис. 4).

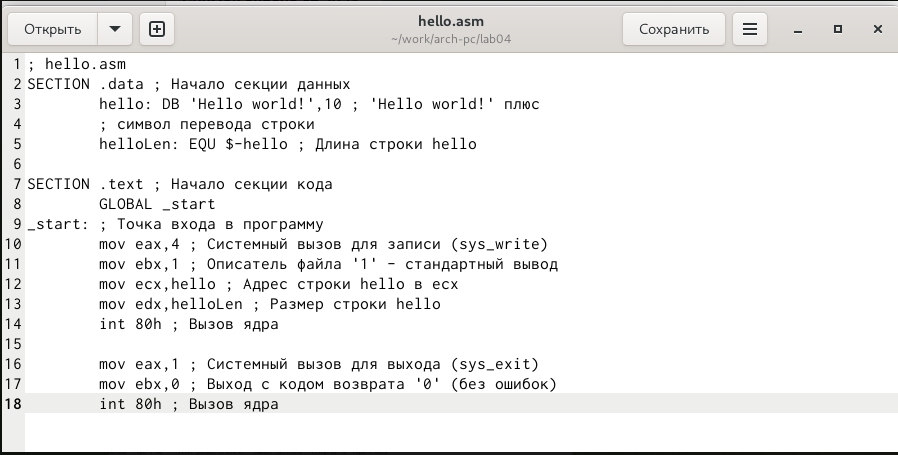


Рис. 4: Заполнение файла

## 3.2 Работа с транслятором NASM

Превращаю 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. Далее проверяю правильность выполнения команды с помощью утилиты ls: действительно, создан файл “hello.o”(рис. 5).

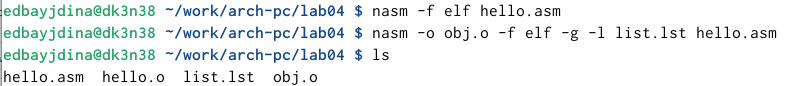


Рис. 5: Компиляция текста программы.png

## 3.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу 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 list.lst. Далее проверяю с помощью утилиты ls правильность выполнения команды.(рис. 6).

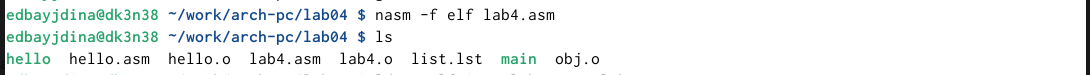


Рис. 6: Компиляция текста программы.png

## 3.4 Работа с компоновщиком LD

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

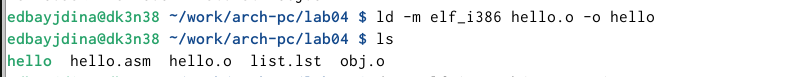


Рис. 7: Передача объектного файла на обработку компоновщику

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

Рис. 8: Передача объектного файла на обработку компоновщику

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

Рис. 9: Запуск исполняемого файла

## 3.6 Выполнение заданий 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

Рис. 10: Создание копии файла

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

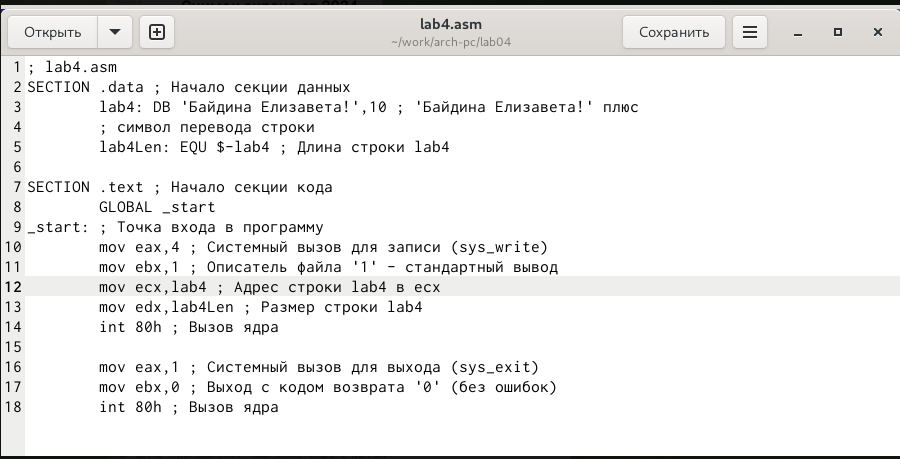


Рис. 11: Изменение программы

Компилирую текст программы в объектный файл. Проверяю с помощью утилиты ls, что файл lab4.o создан (рис. 12).

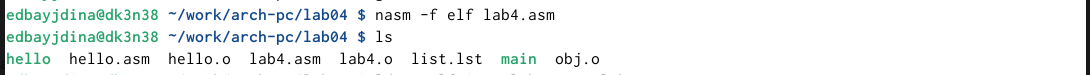


Рис. 12: Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4 (рис. 13).

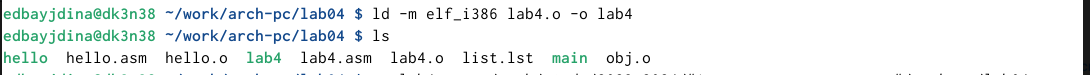


Рис. 13: Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab5, на экран действительно выводятся мои имя и фамилия (рис. 14).

Рис. 14: Запуск исполняемого файла

Рис. 14: Запуск исполняемого файла

Создаю копии файлов в новом каталоге(рис. 15).

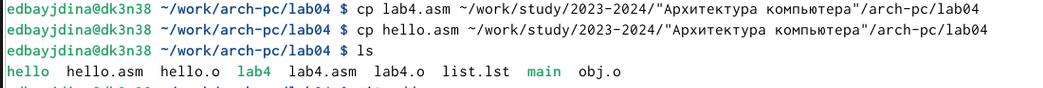


Рис. 15: Создание копии файлов в новом каталоге

С помощью команд git add . и git commit добавляю файлы на GitHub, комментируя действие как добавление файлов для лабораторной работы №4 (рис. 16).

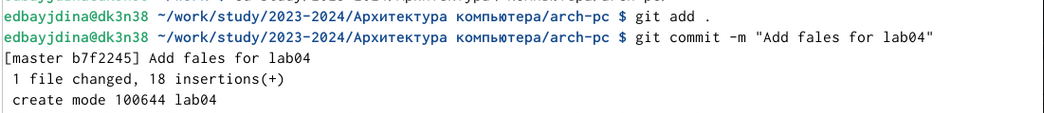


Рис. 16: Добавление файлов на GitHub

Отправляю файлы на сервер с помощью команды git push (рис. 17).

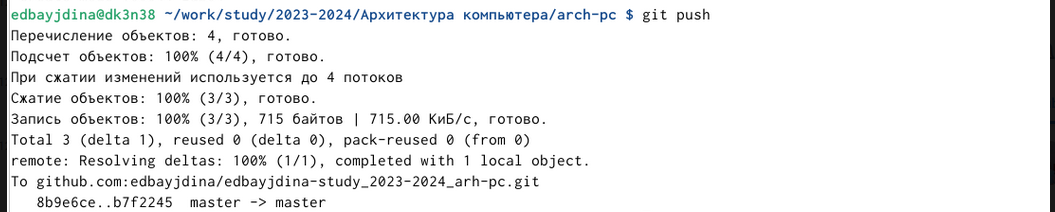


Рис. 17: Отправка файлов

# 4 Выводы

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

# Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.