Лабораторная работа №4

Архитектура вычислительных систем

Абдурахмонов Ихтиёр

Содержание

# 1 Цель работы

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

# 2 Задание

1. В соответствующем каталоге сделайте отчёт по лабораторной работе №4 в формате Markdown. В качестве отчёта необходимо предоставить отчёты в 3 форматах: pdf, docx и md.
2. Загрузите файлы на github.

# 3 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 об Unix см. в [1–6].

# 4 Выполнение лабораторной работы

1. Создаём каталог для работы с программами на языке ассемблера NASM:

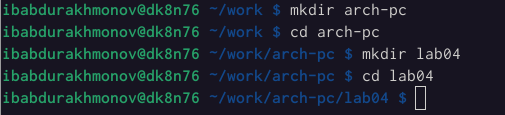


Рис. 1: создание каталога

1. Создаём текстовый файл с именем hello.asm и открываем этот файл с помощью любого текстового редактора gedit:

Рис. 2: gedit

Рис. 2: gedit

1. Вводим в него следующий текст:

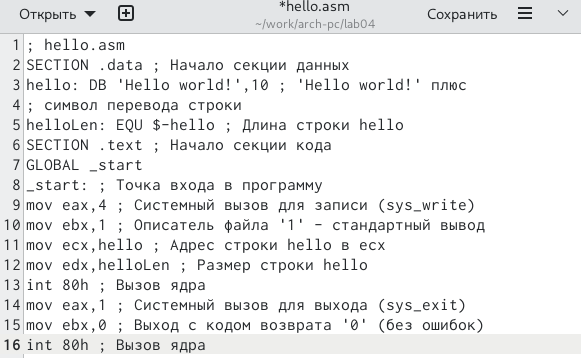


Рис. 3: файл hello.asm

4)NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать следующее

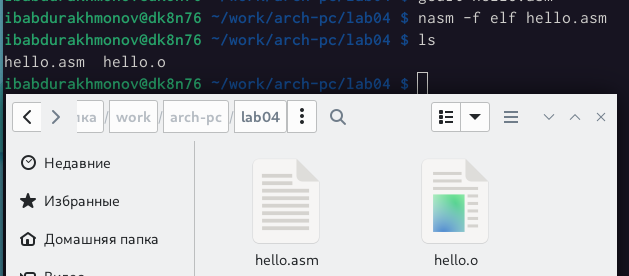


Рис. 4: успешная компиляция

Т. к. текст программы набран без ошибок, транслятор преобразует текст программы из файла hello.asm в объектный код, который записан в файл hello.o.

1. С помощью команды ls проверим, что объектный файл был создан. У нас есть два файла hello.asm и hello.o.

Следующая команда скомпилирует исходный файл hello.asm в obj.o , при этом формат выходного файла будет elf, и в него будут включены символы для отладки (опция -g), кроме того, создается файл листинга list.lst .Выполним следующую команду:

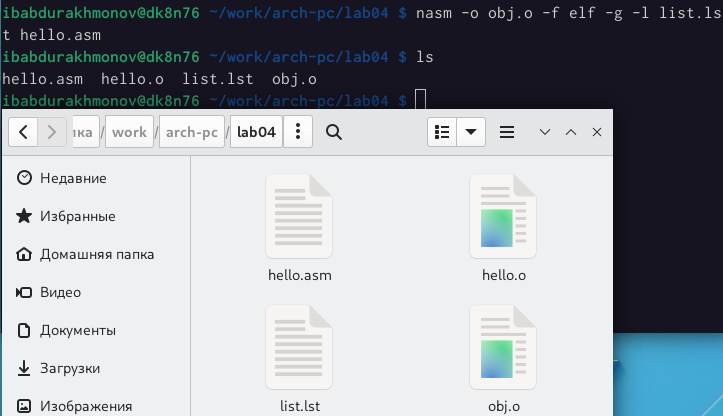


Рис. 5: транслятор

6)Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику, а потом с командой ls проверим содержи- мое:

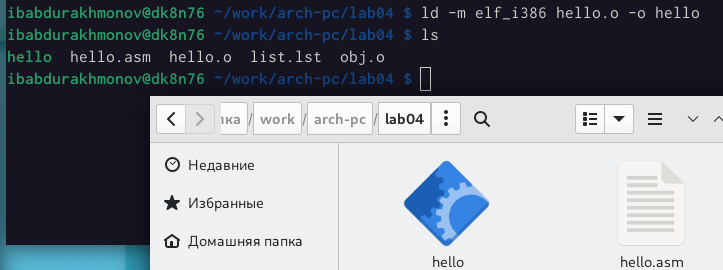


Рис. 6: ged it report.md

1. Ключ -o с последующим значением задаст в данном случае имя создаваемого исполняемого файла. Выполним следующую команду

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику, а потом с командой ls проверим содержимое:

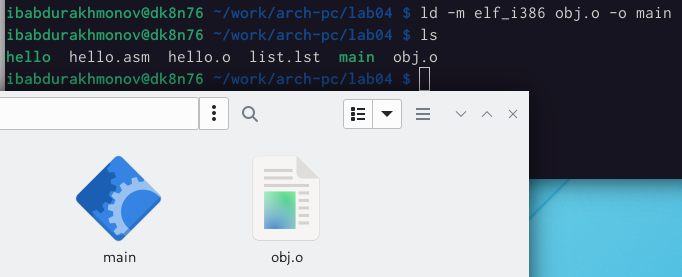


Рис. 7: картинки

11)Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге, набрав в командной строке ./hello:



Рис. 8: файл

# 5 Выполнение самостоятельной работы

1. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создали копию файла hello.asm с именем lab04.asm.

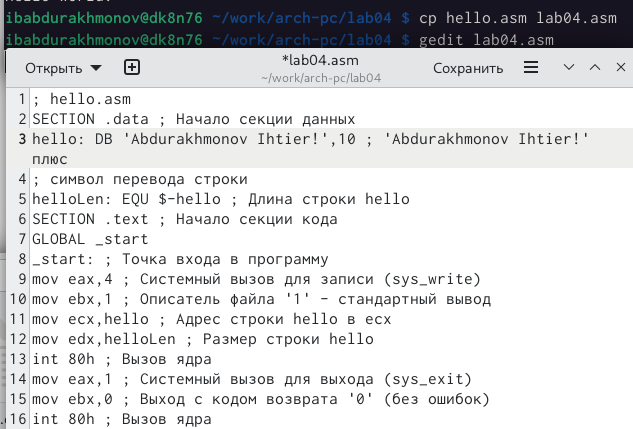


Рис. 9: самостоятельная работа.png

С помощью текстового редактора вносим изменения в текст программы в файле lab04.asm так, чтобы вместо Hello world! на экран выводилась строка с фамилией и именем. Для этого вместо “Hello world” пишем своё имя.

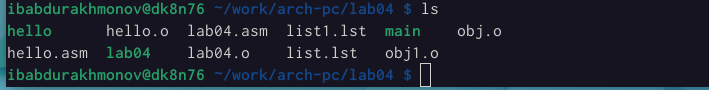


Рис. 10: самостоятельная работа.png

Проводим схожие действия с лабораторной работой, но изменяем название файлов.

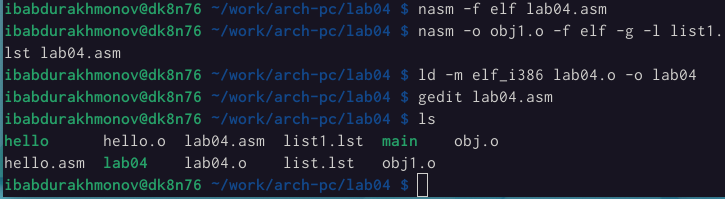


Рис. 11: самостоятельная работа.png

1. Оттранслируем полученный текст программы lab04.asm в объектный файл и запустим, получим вывод фамилии и имени.

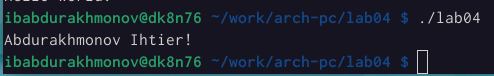


Рис. 12: самостоятельная работа.png

Переносим файлы в основную папку lab04:

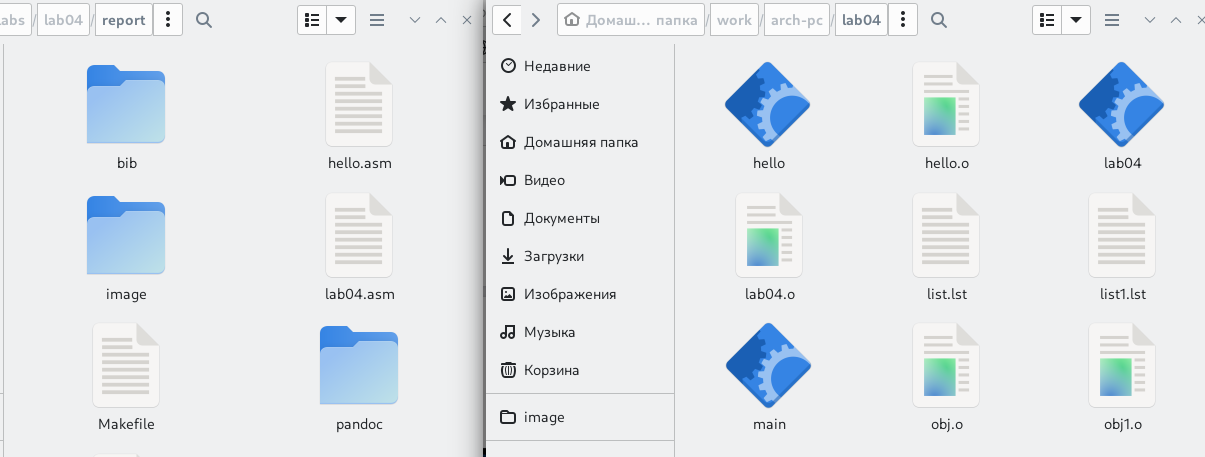


Рис. 13: самостоятельная работа.png

1. Загружаем файлы на GitHub при помощи команд.

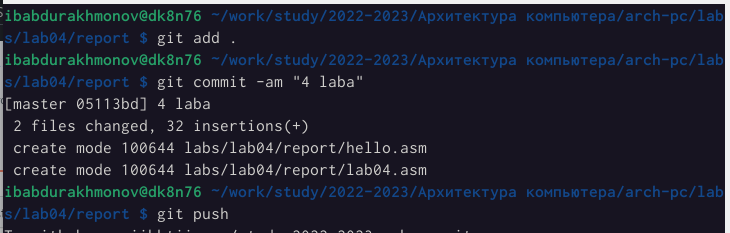


Рис. 14: самостоятельная работа.png

# 6 Выводы

Я освоил процедуру компиляции и сборки программ, написанных на ассемблере NASM.

# Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.

2. 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.

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

4. Robbins A. [Bash Pocket Reference](https://www.ncbi.nlm.nih.gov/pubmed/25246403). O’Reilly Media, 2016. 156 с.

5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.

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