

# Лабораторная работа №4

## Архитектура вычислительных систем

Арсений Сат

### Содержание

#### 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассем- блере 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  
катало

га 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

Имя каталога	Описание каталога
/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:

```
amsat@dk3n31 ~/work $ mkdir ~/work/arch-pc/lab04
amsat@dk3n31 ~/work $ ls
arch-pc  study
```

Рис. 1: создание каталога

- 2) Создаём текстовый файл с именем hello.asm и открываем этот файл с помощью любого текстового редактора gedit

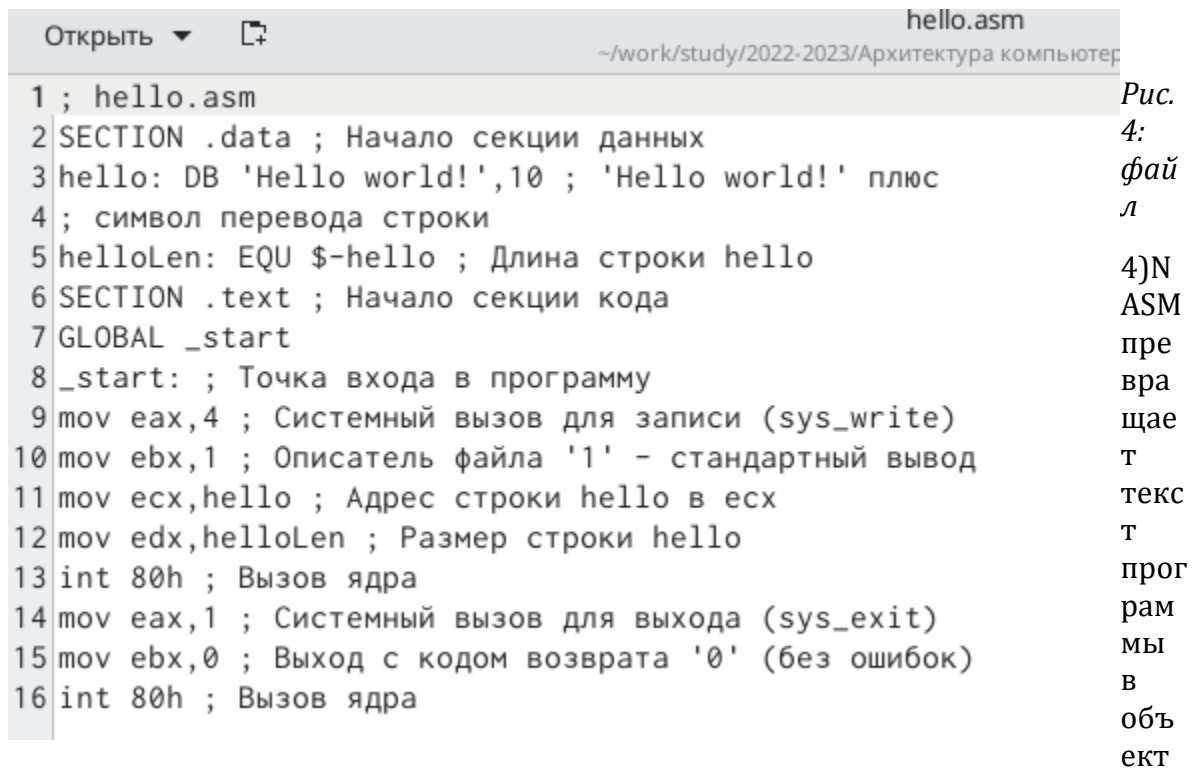
```
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
gedit hello.asm
```

Рис. 2: файл hello.asm

```
amsat@dk2n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
touch hello.asm
```

Рис. 3: gedit

- 3) Вводим в него следующий текст:



ный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать следующее:

```
~/work/arch-pc/lab04 $ nasm -f elf hello.asm
~/work/arch-pc/lab04 $ ls
```

Рис. 5: успешная компиляция

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

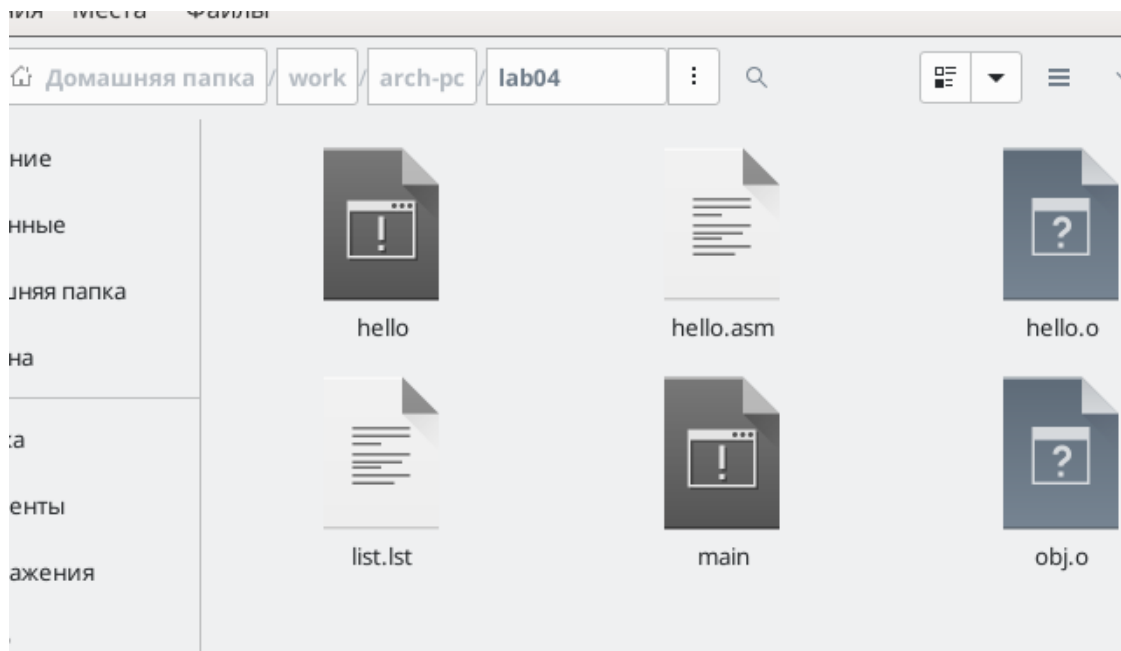


Рис. 6: транслятор

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

```
amsat@dk2n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
ls
hello      hello.o   lab4.o    list.lst  obj1.o    presentation
hello.asm  lab4.asm  list1.lst main       obj.o     report
```

Рис. 7: `ged it report.md`

- 7) 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. 8: картинки

- 8) С помощью команды `ls` проверим, что файлы были созданы:

Рис. 9: `ls`

- 10) Для получения списка форматов объектного файла смотрим `nasm -hf`.

```

amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
nasm -hf
Usage: nasm [-@ response_file] [options...] [--] filename
        nasm -v (or --v)

Options (values in brackets indicate defaults):

    -h                show this text and exit (also --help)
    -v (or --v)       print the NASM version number and exit
    -@ file           response file; one command line option per line

    -o outfile        write output to outfile
    --keep-all        output files will not be removed even if an error happens

    -Xformat          specify error reporting format (gnu or vc)
    -s                redirect error messages to stdout
    -Zfile            redirect error messages to file

    -M                generate Makefile dependencies on stdout
    -MG              d:o, missing files assumed generated
    -MF file          set Makefile dependency file
    -MD file          assemble and generate dependencies
    -MT file          dependency target name
    -MQ file          dependency target name (quoted)

```

Рис. 10: `-hf`

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

Рис. 11: `ls`

12) Ключ `-o` с последующим значением задаст в данном случае имя создаваемого исполняемого файла. Выполним следующую команду:

```

amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
ld -m elf_i386 hello.o -o hello
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
ls
hello  hello.asm  hello.o  list.lst  obj.o  presentation  report

```

```
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
ld -m elf_i386 obj.o -o main
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 12: Ключ -o

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

```
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
./hello
Hello world!
```

Рис. 13: 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.

```
~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ cp hello.asm lab04.asm
```

Рис. 14: 111.png

- 2) С помощью текстового редактора внесли 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" пишу "amsat".

```
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
nasm -f elf lab4.asm
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
ld -m elf_i386 lab4.o -o hello
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
nasm -o obj1.o -f elf -g -l list1.lst lab4.asm
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
./hello
Сат Арсений
```

Рис. 15: 222.png

```
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
./hello
Сат Арсений
```

Рис. 16: 223.png

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

Рис. 17: 333.png

- 4) Загружаю файлы на GitHub при помощи команд "git add .", "git commit -am"" "git push".

```
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
git add .
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
git commit -am 'feat(main): add files lab-4'
[master 23dcfb4] feat(main): add files lab-4
10 files changed, 67 insertions(+)
create mode 100755 labs/lab04/hello
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/hello.o
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/lab4.o
create mode 100644 labs/lab04/list.lst
create mode 100644 labs/lab04/list1.lst
create mode 100755 labs/lab04/main
create mode 100644 labs/lab04/obj.o
create mode 100644 labs/lab04/obj1.o
amsat@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
git push
Перечисление объектов: 17, готово.
Подсчет объектов: 100% (17/17), готово.
При сжатии изменений используется до 4 потоков
```

Рис. 18: 0.png

## 6 Выводы

Я освоила процедуру компиляции и сборки программ, написанных на ассемблере NASM (рассмотрела пример простой программы, научилась изменять внутрисодержимое файла, приобрела навык по созданию объектных файлов).

## 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. O'Reilly Media, 2005. 354 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. 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.