

# **Отчёт по лабораторной работе №4**

**Простейший вариант**

Тимур Ринатович Каримов

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>11</b>
	<b>Список литературы</b>	<b>12</b>

## Список иллюстраций

4.1	Создание каталога и переход в него. . . . .	8
4.2	Создание и открытие текстового файла hello.asm. . . . .	8
4.3	Компиляция текста программы и проверка выполнения команды	8
4.4	Выполнение полного варианта командной строки nasm. . . . .	9
4.5	Передача объектного файла на обработку компоновщику. . . . .	9
4.6	Запуск программы . . . . .	9
4.7	Создание копии файла и переход в него . . . . .	9
4.8	Компиляция текста программы и проверка выполнения команды.	9
4.9	Передача объектного файла на обработку компоновщик и запуск программы. . . . .	10
4.10	Копирование файлов в локальный репозиторий. . . . .	10
4.11	Загрузка файлов на GitHub-1 . . . . .	10
4.12	Загрузка файлов на GitHub-2 . . . . .	10

# Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

# 1 Цель работы

Цель данной лабораторной работы заключается в изучении процесса компиляции и сборки программ, написанных на ассемблере NASM.

## **2 Задание**

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

## 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux	
Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/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].

## 4 Выполнение лабораторной работы

Напишем простую программу на ассемблере, которая выводит приветственное сообщение *Hello, World!*. Создадим каталог для работы с программой и перейдем в него.(рис 4.1)

```
timurkarimov@fedora:~$ mkdir -p ~/work/arch-pc/lab-04  
timurkarimov@fedora:~$ cd ~/work/arch-pc/lab-04
```

Рис. 4.1: Создание каталога и переход в него.

Далее создадим текстовый файл с названием *hello.asm* и откроем его с помощью текстового редактора gedit(рис 4.2)

```
timurkarimov@fedora:~/work/arch-pc/lab-04$ touch hello.asm  
timurkarimov@fedora:~/work/arch-pc/lab-04$ gedit hello.asm
```

Рис. 4.2: Создание и открытие текстового файла hello.asm.

Скомпилируем текст программы *Hello, World!*. Затем посмотрим преобразовали ли транслятор текст программы из файла hello.asm в объектный код, который запишется в файл hello.o с помощью команды ls.(рис. 4.3)

```
timurkarimov@fedora:~/work/arch-pc/lab-04$ nasm -f elf hello.asm  
timurkarimov@fedora:~/work/arch-pc/lab-04$ ls  
hello hello.asm hello.o list.lst main obj.o
```

Рис. 4.3: Компиляция текста программы и проверка выполнения команды

Скомпилируем файл hello.asm, создавая объектный файл obj.o в формате ELF, генерируя отладочную информацию и записывая список ассемблирования в файл list.lst.(рис. 4.4)



```
timurkarimov@fedora:~/work/arch-pc/lab-04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 4.4: Выполнение полного варианта командной строки nasm.

Передадим объектный файл на обработку компоновщику, чтобы получить исполняемую программу. Сделаем так же исполняемый файл и объектный файл с названиями obj.o и main соответственно.(рис. 4.5)

```
timurkarimov@fedora:~/work/arch-pc/lab-04$ ld -m elf_i386 hello.o -o hello
timurkarimov@fedora:~/work/arch-pc/lab-04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
timurkarimov@fedora:~/work/arch-pc/lab-04$ ld -m elf_i386 obj.o -o main
```

Рис. 4.5: Передача объектного файла на обработку компоновщику.

Наконец, запустим исполняемый файл.(рис. 4.6)

```
timurkarimov@fedora:~/work/arch-pc/lab-04$ ./hello
Hello world!
```

Рис. 4.6: Запуск программы

### Приступим к выполнению самостоятельной работы

Создадим копию файла hello.asm с именем lab4.asm в каталоге ~/work/arch-pc/lab04. (рис. 4.7)

```
timurkarimov@fedora:~$ cd ~/work/arch-pc/lab-04
timurkarimov@fedora:~/work/arch-pc/lab-04$ cp hello.asm lab4.asm
timurkarimov@fedora:~/work/arch-pc/lab-04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
```

Рис. 4.7: Создание копии файла и переход в него

Отредактируем с помощью gedit, чтобы вместо *Hello, World!* на экран выводилась моя фамилия с именем. Оттранслируем полученный текст программы lab4.asm в объектный файл.(рис. 4.8)

```
timurkarimov@fedora:~/work/arch-pc/lab-04$ nasm -f elf lab4.asm
timurkarimov@fedora:~/work/arch-pc/lab-04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
timurkarimov@fedora:~/work/arch-pc/lab-04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
```

Рис. 4.8: Компиляция текста программы и проверка выполнения команды.

Выполним компоновку объектного файла и запустим получившийся исполняемый файл.(рис. 4.9)

```
timurkarimov@fedora:~/work/arch-pc/lab-04$ ld -m elf_i386 lab4.o -o lab4
timurkarimov@fedora:~/work/arch-pc/lab-04$ ld -m elf_i386 obj.o -o main
timurkarimov@fedora:~/work/arch-pc/lab-04$ ./lab4
Karimov Timur
```

Рис. 4.9: Передача объектного файла на обработку компоновщик и запуск программы.

Скопируем файлы hello.asm и lab4.asm по ветке ~/work/study/2024-2025/“Архитектура компьютера”/arch-pc/labs/lab04/.(рис. 4.10)

```
timurkarimov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/ab04$ cp ~/work/arch-pc/lab-04/hello.asm hello.asm
timurkarimov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/ab04$ cp ~/work/arch-pc/lab-04/lab4.asm lab4.asm
timurkarimov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/ab04$ ls
hello.asm lab4.asm presentation report
```

Рис. 4.10: Копирование файлов в локальный репозиторий.

Загрузка файлов на GitHub.(рис. 4.11)(рис. 4.12)

```
timurkarimov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/ab04$ git add .
timurkarimov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/ab04$ git commit -am 'feat(main) add files lab-4'
[master f79be781 feat(main) add files lab-4]
```

Рис. 4.11: Загрузка файлов на GitHub-1

```
timurkarimov@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/ab04$ git push
Перечисление объектов: 19, готово.
Подсчет объектов: 100% (19/19), готово.
```

Рис. 4.12: Загрузка файлов на GitHub-2

## 5 Выводы

В ходе лабораторной работы №4, посвященной созданию и обработке программ на языке ассемблера *NASM*, была успешно освоена процедура компиляции и сборки ассемблерных программ. Мы изучили ключевые этапы, включая написание исходного кода, его компиляцию с помощью *NASM* и линковку для получения исполняемого файла.

## 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. O'Reilly Media, 2005. 354 с.