

# **Лабораторная работа №4**

**Создание и процесс обработки программ на языке ассемблера  
NASM**

Акунаева Антонина Эрдниевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Описание результатов выполнения заданий для самостоятельной работы</b>	<b>11</b>
<b>5</b>	<b>Выводы</b>	<b>14</b>

# Список иллюстраций

3.1	Использование команды <code>mkdir -p</code> . . . . .	7
3.2	Использование команды <code>cd</code> . . . . .	7
3.3	Использование команды <code>touch</code> . . . . .	7
3.4	Использование <code>gedit</code> . . . . .	8
3.5	Код ассемблера NASM в <code>gedit</code> . . . . .	8
3.6	Запуск исполняемого файла . . . . .	10
4.1	Использование команды <code>ср</code> . . . . .	11
4.2	Использование <code>gedit</code> . Редакция текста . . . . .	12

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

Вывести Hello World с помощью ассемблера NASM.

Компиляция шаблонов с использованием Makefile.

Удаление шаблонов с использованием Makefile.

Открытие markdown-файла при помощи gedit.

Компиляция отчётов с использованием Makefile и их загрузка на Github.

### 3 Выполнение лабораторной работы

1. Создайте каталог для работы с программами на языке ассемблер NASM.

```
aeakunaeva@fedora:~$ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Использование команды mkdir -p

Содадим каталог lab04 с указанным в тексте лабораторной расположением при помощи команды mkdir и ключа -p для создания иерархической цепочки каталогов и подкаталогов.

2. Перейдите в созданный каталог.

```
aeakunaeva@fedora:~$ cd ~/work/arch-pc/lab04  
aeakunaeva@fedora:~/work/arch-pc/lab04$
```

Рис. 3.2: Использование команды cd

Переходим в каталог при помощи команды cd и указания относительного пути до каталога.

3. Создайте текстовый файл с именем hello.asm.

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ touch hello.asm  
aeakunaeva@fedora:~/work/arch-pc/lab04$ ls  
hello.asm
```

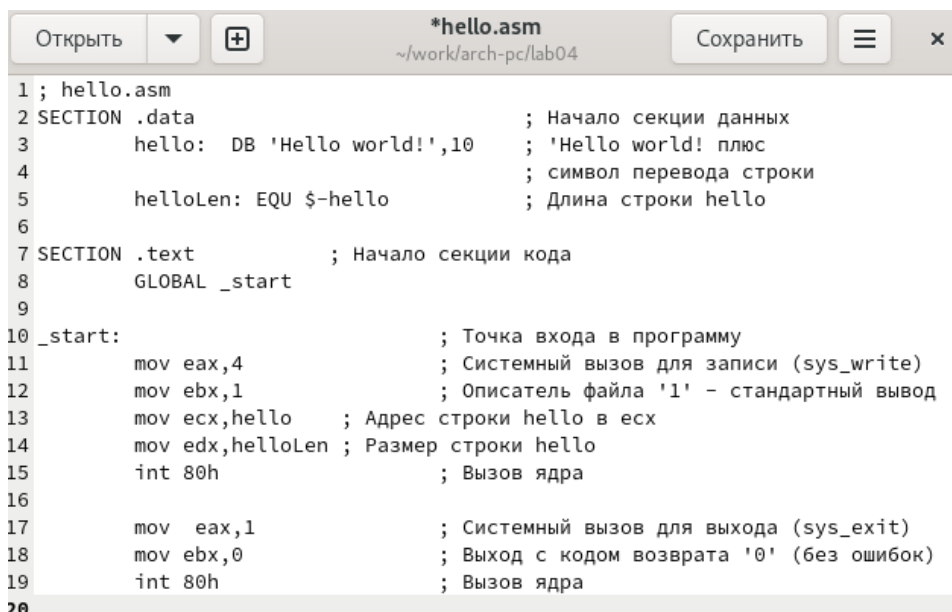
Рис. 3.3: Использование команды touch

Командой `touch` и указанием названия файла и его расширения создадим требуемый файл в новом каталоге. Проверим при помощи `ls`.

4. Откройте этот файл с помощью любого текстового редактора, например, `gedit`, и введите в него следующий текст:

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ gedit hello.asm
```

Рис. 3.4: Использование `gedit`



```
*hello.asm
~/work/arch-pc/lab04

1 ; hello.asm
2 SECTION .data                ; Начало секции данных
3     hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4                                     ; символ перевода строки
5     helloLen: EQU $-hello      ; Длина строки hello
6
7 SECTION .text                ; Начало секции кода
8     GLOBAL _start
9
10 _start:                      ; Точка входа в программу
11     mov eax,4                ; Системный вызов для записи (sys_write)
12     mov ebx,1                ; Описатель файла '1' - стандартный вывод
13     mov ecx,hello            ; Адрес строки hello в ecx
14     mov edx,helloLen         ; Размер строки hello
15     int 80h                  ; Вызов ядра
16
17     mov eax,1                ; Системный вызов для выхода (sys_exit)
18     mov ebx,0                ; Выход с кодом возврата '0' (без ошибок)
19     int 80h                  ; Вызов ядра
20
```

Рис. 3.5: Код ассемблера NASM в `gedit`

При помощи команды `gedit [имя файла]` в текущей директории откроем текстовый редактор, в который вставим текст из лабораторной.

5. Для компиляции приведённого выше текста программы “Hello World” необходимо написать:

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ nasm -f elf hello.asm
aeakunaeva@fedora:~/work/arch-pc/lab04$ ls
hello.asm hello.o
```



6. С помощью команды `ls` проверьте, что объектный файл был создан. Какое имя имеет объектный файл?

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
aeakunaeva@fedora:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
```

Команда `ls` показывает, что объектный файл под названием `hello.o` был создан.

7. Выполните следующую команду: `nasm -o obj.o -f elf -g -l list.lst hello.asm`. С помощью команды `ls` проверьте, что файлы были созданы.

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
aeakunaeva@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

.

8. Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику. С помощью команды `ls` проверьте, что исполняемый файл `hello` был создан.

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
aeakunaeva@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

.

9. Выполните следующую команду: `ld -m elf_i386 obj.o -o main`. Какое имя будет иметь исполняемый файл? Какое имя имеет объектный файл, из которого собран этот исполняемый файл?

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ ./hello
Hello world!
```

.

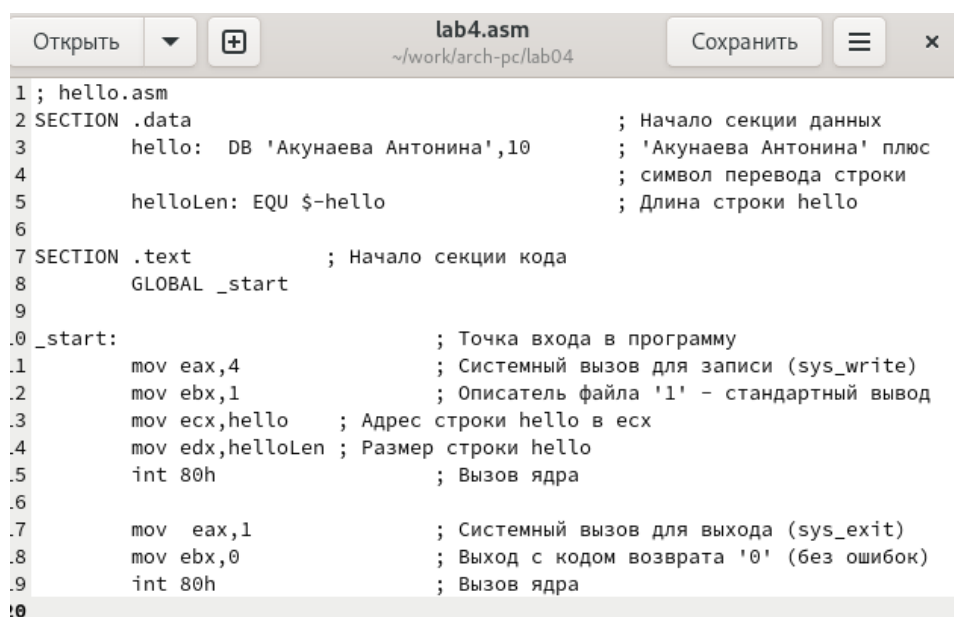
10. Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге можно, набрав в командной строке `./hello`.

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
aeakunaeva@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
```

Рис. 3.6: Запуск исполняемого файла

## 4 Описание результатов выполнения заданий для самостоятельной работы

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm`.



```
1 ; hello.asm
2 SECTION .data                                ; Начало секции данных
3     hello: DB 'Акунаева Антонина',10          ; 'Акунаева Антонина' плюс
4                                           ; символ перевода строки
5     helloLen: EQU $-hello                    ; Длина строки hello
6
7 SECTION .text                                ; Начало секции кода
8     GLOBAL _start
9
10 _start:                                     ; Точка входа в программу
11     mov eax,4                               ; Системный вызов для записи (sys_write)
12     mov ebx,1                               ; Описатель файла '1' - стандартный вывод
13     mov ecx,hello                           ; Адрес строки hello в ecx
14     mov edx,helloLen                        ; Размер строки hello
15     int 80h                                ; Вызов ядра
16
17     mov eax,1                               ; Системный вызов для выхода (sys_exit)
18     mov ebx,0                               ; Выход с кодом возврата '0' (без ошибок)
19     int 80h                                ; Вызов ядра
20
```

Рис. 4.1: Использование команды `cp`

Скопируем файл `hello.asm` с указанием относительных путей до файлов (старое и новое расположение) с изменением названия скопированного файла на `lab4.asm`. Проверим с помощью `ls` корректность выполнения.

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.

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
aeakunaeva@fedora:~/work/arch-pc/lab04$ nasm -o lab4.o -f elf -g lab4.asm
aeakunaeva@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
aeakunaeva@fedora:~/work/arch-pc/lab04$ ./lab4
Акунаева Антонина
```

Рис. 4.2: Использование gedit. Редакция текста

Откроем файл lab4.asm при помощи текстового редактора gedit (gedit lab4.asm) и заменим часть текста с Hello world! на Akunaeva Antonina.

3. Оттранслируйте полученный текст программы lab4.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.

```
aeakunaeva@fedora:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04/
aeakunaeva@fedora:~/work/arch-pc/lab04$ ^C
aeakunaeva@fedora:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04/
aeakunaeva@fedora:~/work/arch-pc/lab04$ ls ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab04/
hello.asm lab4.asm report
```

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.

```

aeakunaeva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
aeakunaeva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am
'feat(main): add files lab-4 test'
[master d3abb89] feat(main): add files lab-4 test
19 files changed, 250 insertions(+)
create mode 100644 labs/Makefile
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/Makefile
create mode 100644 labs/lab04/report/image/1.jpg
create mode 100644 labs/lab04/report/image/10.jpg
create mode 100644 labs/lab04/report/image/11.jpg
create mode 100644 labs/lab04/report/image/12.jpg
create mode 100644 labs/lab04/report/image/13.jpg
create mode 100644 labs/lab04/report/image/14.jpg
create mode 100644 labs/lab04/report/image/2.jpg
create mode 100644 labs/lab04/report/image/3.jpg
create mode 100644 labs/lab04/report/image/4.jpg
create mode 100644 labs/lab04/report/image/5.jpg
create mode 100644 labs/lab04/report/image/6.jpg
create mode 100644 labs/lab04/report/image/7.jpg
create mode 100644 labs/lab04/report/image/8.jpg
create mode 100644 labs/lab04/report/image/9.jpg
create mode 100644 labs/lab04/report/report.md
aeakunaeva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 26, готово.
Подсчет объектов: 100% (26/26), готово.
Сжатие объектов: 100% (24/24), готово.
Запись объектов: 100% (24/24), 235.75 КиБ | 2.18 МиБ/с, готово.
Total 24 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:axelxi/study_2024-2025_arh-pc.git
   337b06b..d3abb89  master -> master
aeakunaeva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ 

```

## 5 Выводы

Я освоила процедуры оформления отчетов с помощью легковесного языка разметки Markdown.