

# **Отчет по лабораторной работе №5**

Смирнов Дмитрий Романович, НММбд-03-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
<b>3</b>	<b>Задания для самостоятельной работы</b>	<b>8</b>
<b>4</b>	<b>Вопросы для самопроверки</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>11</b>

# Список иллюстраций

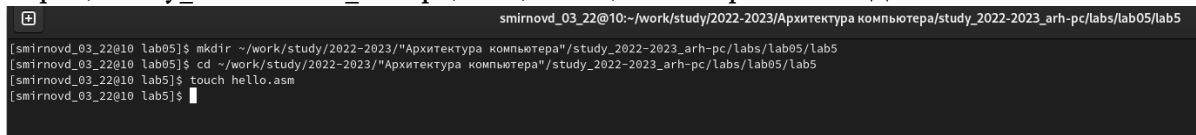
2.1	Рис2 . . . . .	5
2.2	Рис3 . . . . .	6
2.3	Рис4 . . . . .	6
2.4	Рис5 . . . . .	6
2.5	Рис6 . . . . .	6
2.6	Рис7 . . . . .	7
3.1	Рис8 . . . . .	8
3.2	Рис9 . . . . .	8

# 1 Цель работы

Целью работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

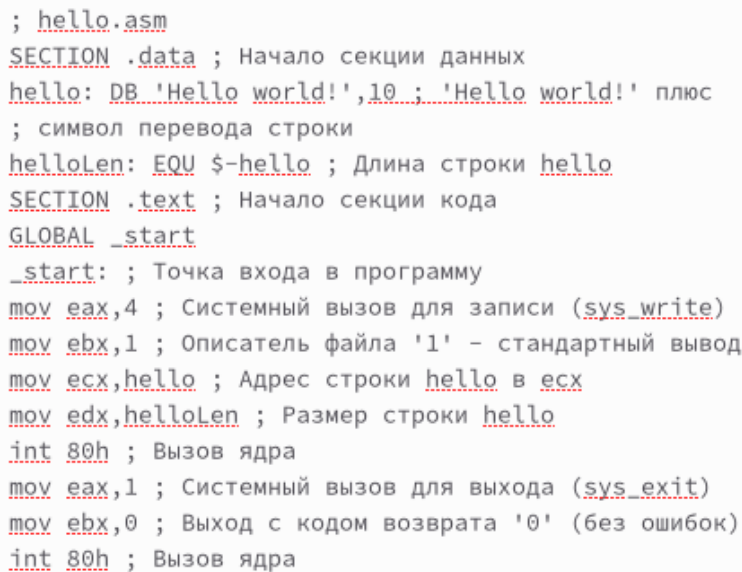
## 2 Выполнение лабораторной работы

Создам в каталоге `cd ~/work/study/2022-2023/“Архитектура компьютера”/study_2022-2023_arh-pc/labs/lab05/lab5` файл под названием `hello`



```
smirnovd_03_22@10:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab5]$ touch hello.asm
[smirnovd_03_22@10 lab5]$
```

Открою и напишу в этом текстовом файле программу выводящую “Hello world”



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

Рис. 2.1: Рис2

Превращу текст программы в объектный код

```
smirnovd_03_22@10:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab5]$ touch hello.asm
[smirnovd_03_22@10 lab5]$ gedit hello.asm
[smirnovd_03_22@10 lab5]$ nasm -f elf hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o
[smirnovd_03_22@10 lab5]$
```

Рис. 2.2: Рис3

Превращу исходный код в объектный код с другим названием

```
smirnovd_03_22@10:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab5]$ touch hello.asm
[smirnovd_03_22@10 lab5]$ gedit hello.asm
[smirnovd_03_22@10 lab5]$ nasm -f elf hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o
[smirnovd_03_22@10 lab5]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o  list.lst  obj.o
[smirnovd_03_22@10 lab5]$
```

Рис. 2.3: Рис4

Создам исполняемый файл

```
smirnovd_03_22@10:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab5]$ touch hello.asm
[smirnovd_03_22@10 lab5]$ gedit hello.asm
[smirnovd_03_22@10 lab5]$ nasm -f elf hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o
[smirnovd_03_22@10 lab5]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o  list.lst  obj.o
[smirnovd_03_22@10 lab5]$ ld -m elf_i386 hello.o -o hello
[smirnovd_03_22@10 lab5]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
[smirnovd_03_22@10 lab5]$
```

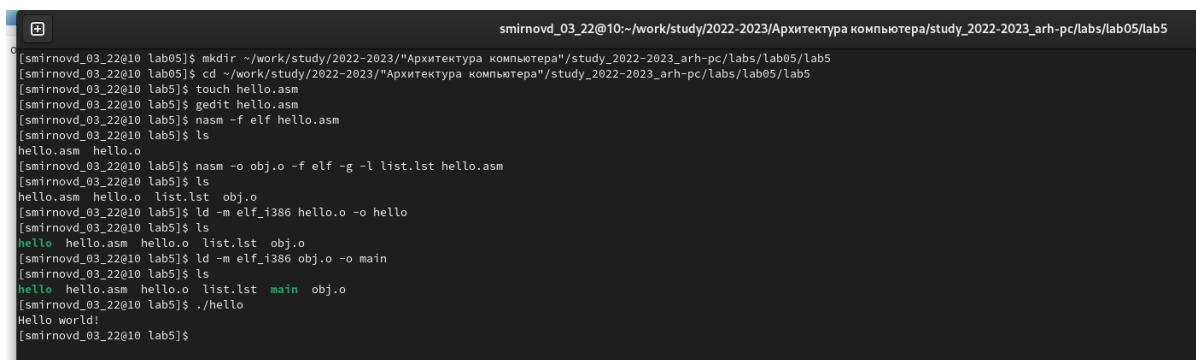
Рис. 2.4: Рис5

Создадим исполняемый файл с другим названием

```
smirnovd_03_22@10:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab5]$ touch hello.asm
[smirnovd_03_22@10 lab5]$ gedit hello.asm
[smirnovd_03_22@10 lab5]$ nasm -f elf hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o
[smirnovd_03_22@10 lab5]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o  list.lst  obj.o
[smirnovd_03_22@10 lab5]$ ld -m elf_i386 hello.o -o hello
[smirnovd_03_22@10 lab5]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
[smirnovd_03_22@10 lab5]$ ld -m elf_i386 obj.o -o main
[smirnovd_03_22@10 lab5]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
[smirnovd_03_22@10 lab5]$
```

Рис. 2.5: Рис6

## Запущу на выполнение созданный исполняемый файл

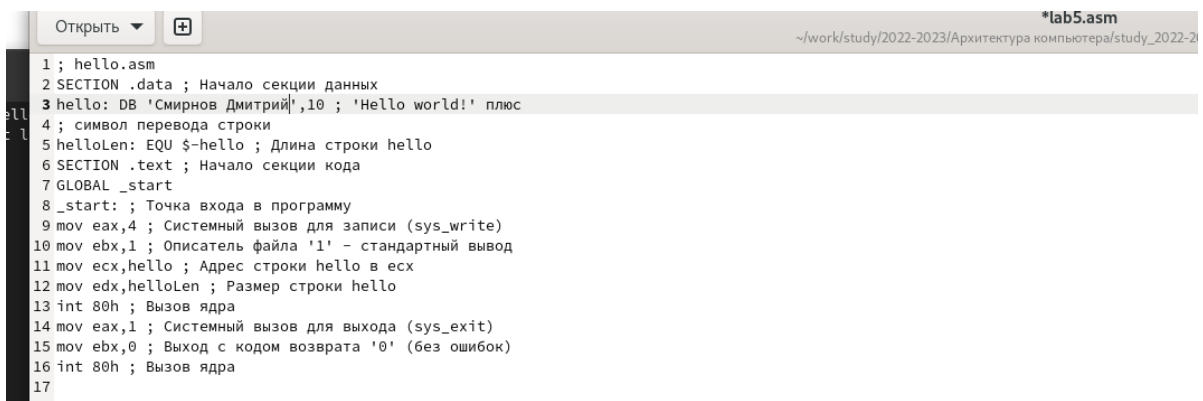


```
smirnovd_03_22@10:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab05]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab5]$ touch hello.asm
[smirnovd_03_22@10 lab5]$ gedit hello.asm
[smirnovd_03_22@10 lab5]$ nasm -f elf hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o
[smirnovd_03_22@10 lab5]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[smirnovd_03_22@10 lab5]$ ls
hello.asm  hello.o  list.lst  obj.o
[smirnovd_03_22@10 lab5]$ ld -m elf_i386 hello.o -o hello
[smirnovd_03_22@10 lab5]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
[smirnovd_03_22@10 lab5]$ ld -m elf_i386 obj.o -o main
[smirnovd_03_22@10 lab5]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
[smirnovd_03_22@10 lab5]$ ./hello
Hello world!
[smirnovd_03_22@10 lab5]$
```

Рис. 2.6: Рис7

### 3 Задания для самостоятельной работы

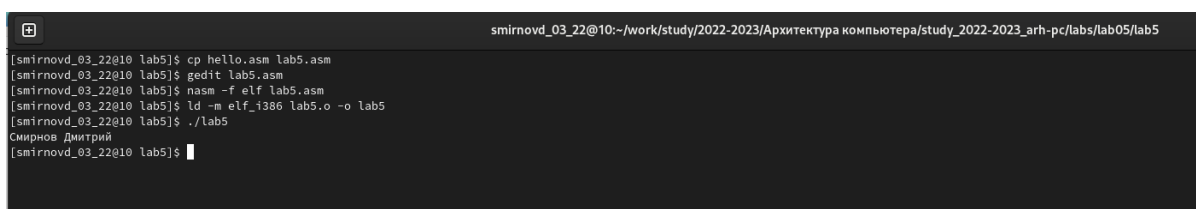
Создадим копию файла hello.asm с названием lab5.asm Внесу изменения в файл lab5.asm так, чтобы на вывод подавались мое имя и фамилия



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

Рис. 3.1: Рис8

Превращу файл lab5.asm в объектный, выполню компоновку и запущу получившийся файл



```
smirnovd_03_22@10:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab05/lab5
[smirnovd_03_22@10 lab5]$ cp hello.asm lab5.asm
[smirnovd_03_22@10 lab5]$ gedit lab5.asm
[smirnovd_03_22@10 lab5]$ nasm -f elf lab5.asm
[smirnovd_03_22@10 lab5]$ ld -m elf_i386 lab5.o -o lab5
[smirnovd_03_22@10 lab5]$ ./lab5
Смирнов Дмитрий
[smirnovd_03_22@10 lab5]$
```

Рис. 3.2: Рис9



## 4 Вопросы для самопроверки

1. Какие основные отличия ассемблерных программ от программ на языках высокого уровня? Ассемблерная программа содержит только тот код, который ввел программист.
2. В чём состоит отличие инструкции от директивы на языке ассемблера? Инструкция непосредственно преобразуется в то, что может выполнить процессор. Директива-это то, что ассемблер может интерпретировать и что-то говорит о том, как должны быть собраны инструкции.
3. Перечислите основные правила оформления программ на языке ассемблера. 1.Порядок 2.Подходы при написании комментариев 3.Проблема должна быть хорошо продумана перед тем, как она сможет быть решена 4.Оптимизация программ на языке ассемблера
4. Каковы этапы получения исполняемого файла? Текст программы, трансляция. компоновка, запуск программы
5. Каково назначение этапа трансляции? Трансляция — преобразование с помощью транслятора, например `nasm`, текста программы в машинный код, называемый объектным. На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором. Тип объектного файла — `o`, файла листинга — `lst`
6. Каково назначение этапа компоновки? Компоновка — этап обработки объектного кода компоновщиком (`ld`), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно

не имеет расширения. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение map.

7. Какие файлы могут создаваться при трансляции программы, какие из них создаются по умолчанию? Объектные файлы - по умолчанию, а листинг - нет
8. Каковы форматы файлов для nasm и ld? NASM - поддерживает широкий диапазон форматов объектных файлов, включая форматы Linux.out и ELF, NetBSD/FreeBSD, COFF, Microsoft 16-bit OBJ и Win32. LD - elf64-x86-64 elf32-i386 elf32-iamcu elf32-x86-64 pei-i386 pe-x86-64 pei-x86-64 elf64-l1om elf64-k1om elf64-little elf64-big elf32-little elf32-big pe-bigobj-x86-64 pe-i386 elf64-bpfile elf64-bpfbe srec symbolsrec verilog tekhex binary ihex plugin ld

## 5 Выводы

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