

# **Отчет по лабораторной работа №5**

**Группа НПИбд-02-22**

Стариков Данила Андреевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>3</b>
<b>2</b>	<b>Основная часть</b>	<b>4</b>
2.1	Выполнение лабораторной работы . . . . .	4
2.2	Выполнение заданий для самостоятельной работы. . . . .	6
<b>3</b>	<b>Выводы</b>	<b>8</b>

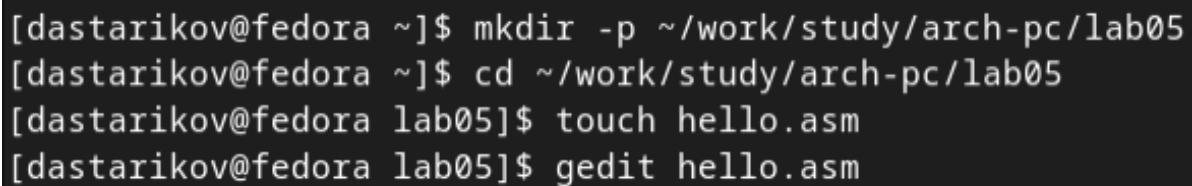
# 1 Цель работы

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

## 2 Основная часть

### 2.1 Выполнение лабораторной работы

Для изучения работы с программами на языке ассемблера NASM в качестве примера взяли программу, выводящую сообщение “Hello, world!” на экран. В каталоге курса создали подкаталог для работы с программами, в нем – файл hello.asm, содержащий текст программы (Рис. 2.1, 2.2).

A screenshot of a terminal window showing four commands being executed in sequence. The first command creates a directory, the second changes to that directory, the third creates a new file, and the fourth opens the file in a text editor.

```
[dastarikov@fedora ~]$ mkdir -p ~/work/study/arch-pc/lab05  
[dastarikov@fedora ~]$ cd ~/work/study/arch-pc/lab05  
[dastarikov@fedora lab05]$ touch hello.asm  
[dastarikov@fedora lab05]$ gedit hello.asm
```

Рис. 2.1: Создание файла hello.asm.

```

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 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

```

Рис. 2.2: Текст программы hello.asm.

После создания файла с кодом программы транслируем его в объектный файл hello.o (Рисунок 2.3). Команду nasm можно запустить с дополнительными параметрами: исходный файл hello.asm скомпилируется в obj.o (опция -o позволяет задать имя объектного файла, в данном случае obj.o), при этом формат выходного файла будет elf, и в него будут включены символы для отладки (опция -g), кроме того, будет создан файл листинга list.lst (опция -l) (Рисунок 2.4).

```

[dastarikov@fedora lab05]$ nasm -f elf hello.asm
[dastarikov@fedora lab05]$ ls
hello.asm  hello.o

```

Рис. 2.3: Трансляция файла hello.asm.

```
[dastarikov@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[dastarikov@fedora lab05]$ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 2.4: Трансляция файла hello.asm. с дополнительными параметрами.

На последнем этапе объектный файл компоуем с помощью компоновщика ld (Рисунок 2.5). Ключ -o дает возможность переименовать получаемый исполняемый файл (Рисунок 2.6).

```
[dastarikov@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[dastarikov@fedora lab05]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 2.5: Компоновка объектного файла hello.o.

```
[dastarikov@fedora lab05]$ ld -m elf_i386 obj.o -o main
[dastarikov@fedora lab05]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 2.6: Компоновка объектного файла obj.o и его переименование в main

В результате компоновки получили исполняемый файл hello без расширения, при запуске получаем сообщение “Hello, world!” (Рисунок 2.7).

```
[dastarikov@fedora lab05]$ ./hello
Hello world!
[dastarikov@fedora lab05]$
```

Рис. 2.7: Демонстрация работы программы lab05.

## 2.2 Выполнение заданий для самостоятельной работы.

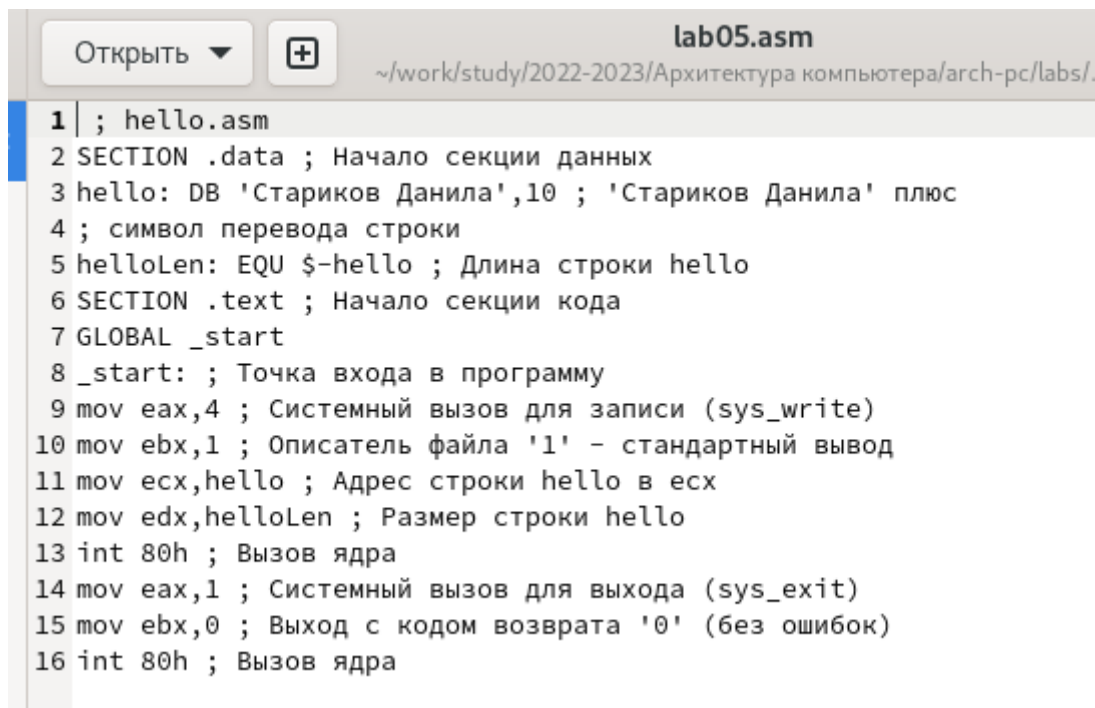
В том же каталоге создали копию файла hello.asm с именем lab05.asm и изменили текст программы, чтобы выводилось сообщение “Стариков Данила” (Рис. 2.8, 2.9).

```

[dastarikov@fedora lab05]$ cp hello.asm lab05.asm
[dastarikov@fedora lab05]$ ;s
bash: синтаксическая ошибка рядом с неожиданным маркером «;»
[dastarikov@fedora lab05]$ ls
hello  hello.asm  hello.o  lab05.asm  list.lst  main  obj.o
[dastarikov@fedora lab05]$ gedit lab05.asm
[dastarikov@fedora lab05]$ nasm -f elf lab05.asm
[dastarikov@fedora lab05]$ ld -m elf_i386 lab05.o -o lab05
[dastarikov@fedora lab05]$ ./lab05
Стариков Данила
[dastarikov@fedora lab05]$ gedit
[dastarikov@fedora lab05]$ ls
hello  hello.asm  hello.o  lab05  lab05.asm  lab05.o  list.lst  main  obj.o
[dastarikov@fedora lab05]$

```

Рис. 2.8: Создание исполняемой программы lab05 и демонстрация ее работы.



```

1 | ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Стариков Данила',10 ; 'Стариков Данила' 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

```

Рис. 2.9: Код программы lab05.asm.

Созданные файлы hello.asm и lab05.asm скопировали в каталог ~/work/study/2022-2023/“Архитектура компьютера”/archpc/labs/lab05/ и загрузили на Github.

## 3 Выводы

В рамках лабораторной работы на примере программы, печатающей “Hello, world!” в консоль, освоили процедуры компиляции и сборки программ, написанных на ассемблере NASM на примере программы.