

# **ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №5**

*дисциплина: Архитектура компьютера*

Лихтенштейн Алина Алексеевна

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выполнение задания для самостоятельной работы	8
4	Выводы	10

## Список иллюстраций

2.1	Новый каталог для работы с программами на языке ассемблера NASM	5
2.2	Создание и редактирование файла <code>hello.asm</code>	5
2.3	<code>hello.asm</code> в текстовом редакторе <code>gedit</code>	5
2.4	Генерация объектного файла	6
2.5	Генерация объектного файла с символами для отладки и файла листинга	6
2.6	Генерация исполняемого файла с именем по умолчанию	6
2.7	Генерация исполняемого файла с заданным именем с помощью объектного файла <code>obj.o</code>	6
2.8	Запуск исполняемой программы	7
3.1	Создание копии файла <code>hello.asm</code>	8
3.2	Команда для запуска файла <code>lab5.asm</code> в текстовом редакторе <code>gedit</code>	8
3.3	<code>lab5.asm</code> в текстовом редакторе <code>gedit</code>	8
3.4	Трансляция исходного текста программы, компоновка объектного файла и запуск получившегося исполняемого файла	9
3.5	Копирование файлов <code>hello.asm</code> и <code>lab5.asm</code> в локальный репозиторий	9

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Создадим каталог для работы с программами на языке ассемблера NASM и перейдем в него. (рис. 2.1)

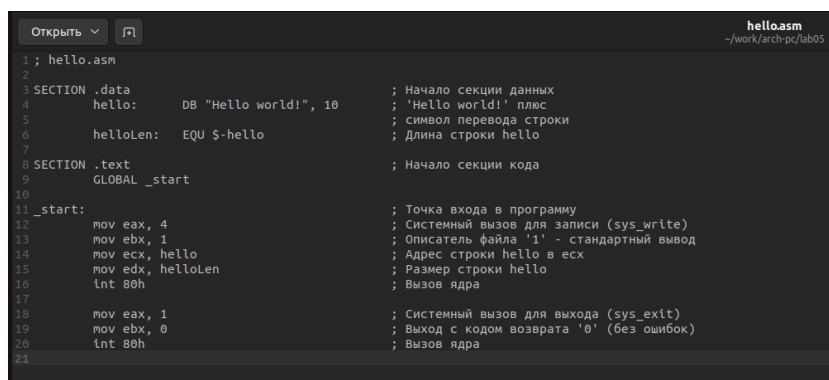
```
aaliechtenstein@rudn:~$ mkdir -p ~/work/arch-pc/lab05
aaliechtenstein@rudn:~$ cd work/arch-pc/lab05/
aaliechtenstein@rudn:~/work/arch-pc/lab05$
```

Рис. 2.1: Новый каталог для работы с программами на языке ассемблера NASM

Создадим текстовый файл с именем hello.asm и откроем его с помощью текстового редактора gedit. (рис. 2.2, 2.3)

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ touch hello.asm
aaliechtenstein@rudn:~/work/arch-pc/lab05$ gedit hello.asm
```

Рис. 2.2: Создание и редактирование файла hello.asm



```
1; hello.asm
2
3SECTION .data
4    hello:    DB "Hello world!", 10    ; Начало секции данных
5                                           ; 'Hello world!' плюс
6    helloLen: EQU $-hello             ; символ перевода строки
7                                           ; Длина строки hello
8SECTION .text
9    GLOBAL _start                     ; Начало секции кода
10
11_start:
12    mov eax, 4                       ; Точка входа в программу
13    mov ebx, 1                       ; Системный вызов для записи (sys_write)
14    mov ecx, hello                   ; Описатель файла '1' - стандартный вывод
15    mov edx, helloLen               ; Адрес строки hello в ecx
16    int 80h                         ; Размер строки hello
17                                           ; Вызов ядра
18    mov eax, 1                       ; Системный вызов для выхода (sys_exit)
19    mov ebx, 0                       ; Выход с кодом возврата '0' (без ошибок)
20    int 80h                         ; Вызов ядра
21
```

Рис. 2.3: hello.asm в текстовом редакторе gedit

Сгенерируем объектный файл исходного текста программы hello.asm. (рис. 2.4)

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ nasm -f elf hello.asm
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ls
hello.asm  hello.o
```

Рис. 2.4: Генерация объектного файла

Имя объектного файла - hello.o - такое же, как и у исходного текстового файла с кодом ассемблера.

Выполним команду, которая 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, будет создан файл листинга list.lst (опция-l). (рис. 2.5)

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ nasm -o obj.o -f elf -g -l list.lst hello.asm
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 2.5: Генерация объектного файла с символами для отладки и файла листинга

Передадим файл на обработку компоновщику, чтобы получить исполняемую программу. (рис. 2.6)

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ld -m elf_i386 hello.o -o hello
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 2.6: Генерация исполняемого файла с именем по умолчанию

Сгенерируем исполняемую программу с помощью объектного файла obj.o, имеющего отличное от исходного текста программы hello.asm название, и дадим программе имя main. (рис. 2.7)

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ld -m elf_i386 obj.o -o main
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 2.7: Генерация исполняемого файла с заданным именем с помощью объектного файла obj.o

Т.о. имя исполняемого файла - main, а имя объектного файла, из которого собран исполняемый файл, - obj.o.

Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге. (рис. 2.8)

A terminal window with a dark background. The prompt is 'aaliechtenstein@rudn:~/work/arch-pc/lab05\$' in green. The command './hello' is entered in white. The output 'Hello world!' is displayed in white on the next line.

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ./hello
Hello world!
```

Рис. 2.8: Запуск исполняемой программы

### 3 Выполнение задания для самостоятельной работы

В каталоге `~/work/arch-pc/lab05` с помощью команды `cp` создадим копию файла `hello.asm` с именем `lab5.asm` (рис. 3.1)

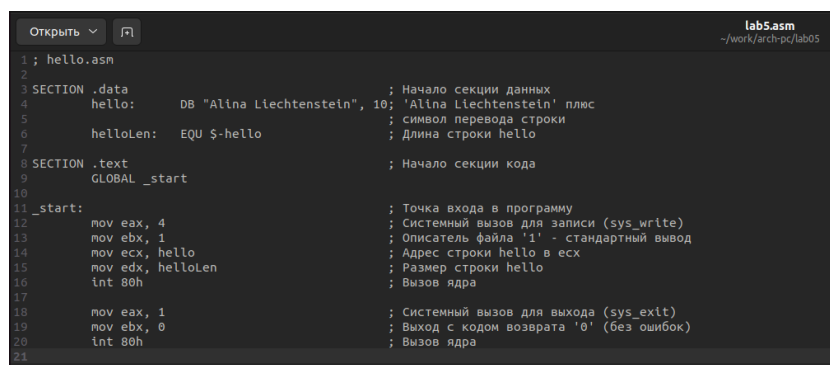
```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ cp hello.asm lab5.asm
aaliechtenstein@rudn:~/work/arch-pc/lab05$
```

Рис. 3.1: Создание копии файла `hello.asm`

С помощью текстового редактора `gedit` внесем изменения в текст программы в файле `lab5.asm` так, чтобы вместо `Hello world!` на экран выводились имя и фамилия студента, выполняющего отчет. (рис. 3.2, 3.3)

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ gedit lab5.asm
```

Рис. 3.2: Команда для запуска файла `lab5.asm` в текстовом редакторе `gedit`



```
Открыть  [icon] lab5.asm
~/work/arch-pc/lab05
1; hello.asm
2
3SECTION .data ; Начало секции данных
4    hello:    DB "Alina Liechtenstein", 10; 'Alina Liechtenstein' плюс
5              ; символ перевода строки
6    helloLen: EQU $-hello ; Длина строки hello
7
8SECTION .text ; Начало секции кода
9    GLOBAL _start
10
11_start: ; Точка входа в программу
12    mov eax, 4 ; Системный вызов для записи (sys_write)
13    mov ebx, 1 ; Описатель файла '1' - стандартный вывод
14    mov ecx, hello ; Адрес строки hello в ecx
15    mov edx, helloLen ; Размер строки hello
16    int 80h ; Вызов ядра
17
18    mov eax, 1 ; Системный вызов для выхода (sys_exit)
19    mov ebx, 0 ; Выход с кодом возврата '0' (без ошибок)
20    int 80h ; Вызов ядра
21
```

Рис. 3.3: `lab5.asm` в текстовом редакторе `gedit`



Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. (рис. 3.4)

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ nasm -f elf lab5.asm
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ld -m elf_i386 lab5.o -o lab5
aaliechtenstein@rudn:~/work/arch-pc/lab05$ ./lab5
Alina Liechtenstein
```

Рис. 3.4: Трансляция исходного текста программы, компоновка объектного файла и запуск получившегося исполняемого файла

Скопируем файлы hello.asm и lab5.asm в локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/archpc/labs/lab05/. (рис. 3.5)

```
aaliechtenstein@rudn:~/work/arch-pc/lab05$ cp hello.asm ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/labs/lab05/hello.asm
aaliechtenstein@rudn:~/work/arch-pc/lab05$ cp lab5.asm ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/labs/lab05/lab5.asm
```

Рис. 3.5: Копирование файлов hello.asm и lab5.asm в локальный репозиторий

## **4 Выводы**

В процессе выполнения лабораторной работы были получены практические навыки, освоены процедуры компиляции и сборки программ, написанных на ассемблере NASM.