

# **Отчёт по лабораторной работе №6**

**Дисциплина: архитектура компьютера**

Сидельников Андрей Владимирович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Выполнение арифметических операций в NASM . . . . .	10
2.2	Ответы на вопросы . . . . .	14
2.3	Выполнение заданий для самостоятельной работы . . . . .	15
<b>3</b>	<b>Выводы</b>	<b>18</b>

## **Список таблиц**

# Список иллюстраций

2.1	Создание директории и файла . . . . .	6
2.2	Копирование файла . . . . .	6
2.3	Редактирование файла . . . . .	7
2.4	Запуск исполняемого файла . . . . .	7
2.5	Редактирование файла . . . . .	7
2.6	Запуск исполняемого файла . . . . .	8
2.7	Создание файла . . . . .	8
2.8	Редактирование файла . . . . .	8
2.9	Запуск исполняемого файла . . . . .	9
2.10	Редактирование файла . . . . .	9
2.11	Запуск исполняемого файла . . . . .	9
2.12	Редактирование файла . . . . .	10
2.13	Запуск исполняемого файла . . . . .	10
2.14	Создание файла . . . . .	10
2.15	Редактирование файла . . . . .	11
2.16	Запуск исполняемого файла . . . . .	11
2.17	Изменение программы . . . . .	12
2.18	Запуск исполняемого файла . . . . .	12
2.19	Создание файла . . . . .	13
2.20	Редактирование файла . . . . .	13
2.21	Запуск исполняемого файла . . . . .	14
2.22	Создание файла . . . . .	15
2.23	Написание программы . . . . .	15
2.24	Запуск исполняемого файла . . . . .	16

# 1 Цель работы

Цель лабораторной работы - освоение арифметических инструкций языка ассемблера NASM

## 2 Выполнение лабораторной работы

Создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6, перехожу в неё и создаю файл lab6-1.asm (рис. 2.1).

```
avsideljnikov@avsideljnikov-VirtualBox:~$ mkdir ~/work/arch-pc/lab06
avsideljnikov@avsideljnikov-VirtualBox:~$ cd ~/work/arch-pc/lab06
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ touch lab6-1.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$
```

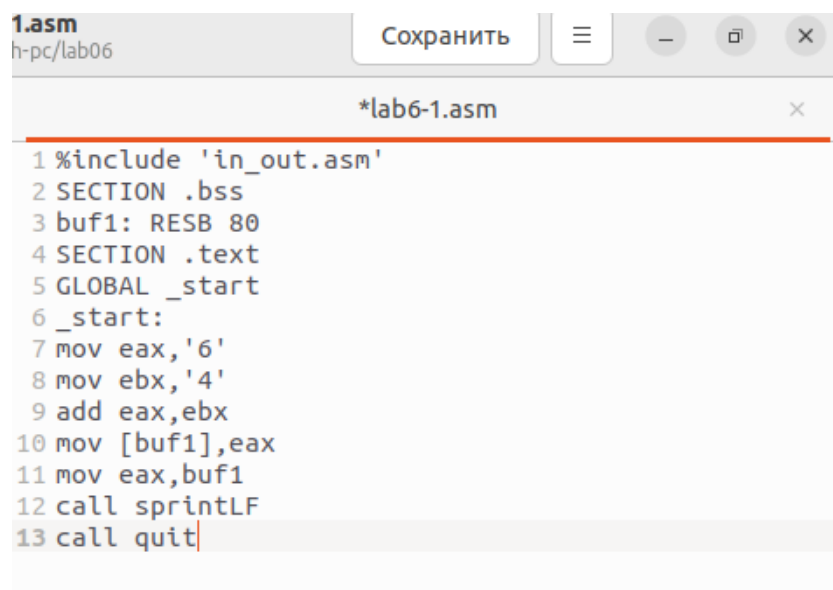
Рис. 2.1: Создание директории и файла

Копирую в текущий каталог файл in\_out.asm с помощью утилиты cp, т.к. он будет использоваться в других программах (рис. 2.2).

```
avsideljnikov@avsideljnikov-VirtualBox:~$ cp /home/avsideljnikov/Загрузки/in_out.asm ~/work/arch-pc/lab06
avsideljnikov@avsideljnikov-VirtualBox:~$ cd ~/work/arch-pc/lab06
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.2: Копирование файла

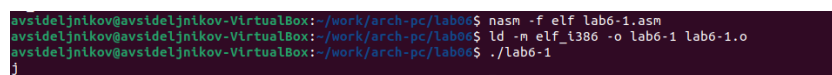
Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax (рис. 2.3).



```
1.asm
h-pc/lab06
Сохранить
*lab6-1.asm
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,'6'
8 mov ebx,'4'
9 add eax,ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintLF
13 call quit
```

Рис. 2.3: Редактирование файла

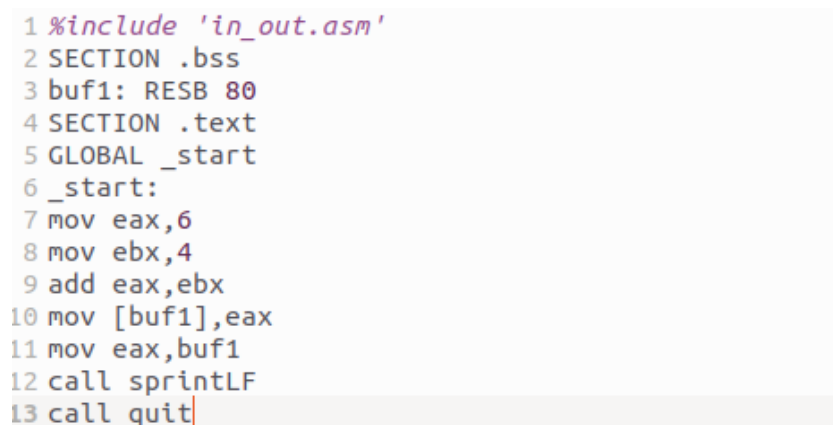
Создаю исполняемый файл программы и запускаю его (рис. 2.4).



```
avsideljnikov@avsideljnikov-VirtualBox: ~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
avsideljnikov@avsideljnikov-VirtualBox: ~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
avsideljnikov@avsideljnikov-VirtualBox: ~/work/arch-pc/lab06$ ./lab6-1
```

Рис. 2.4: Запуск исполняемого файла

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 2.5).



```
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,6
8 mov ebx,4
9 add eax,ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintLF
13 call quit
```

Рис. 2.5: Редактирование файла

Создаю новый исполняемый файл программы и запускаю его, теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран (рис. 2.6).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-1
```

Рис. 2.6: Запуск исполняемого файла

Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. 2.7).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ touch lab6-2.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.7: Создание файла

Ввожу в файл текст программы из листинга 6.2 (рис. 2.8).

```
*lab6-2.asm
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,'6'
6 mov ebx,'4'
7 add eax,ebx
8 call iprintLF
9 call quit
```

Рис. 2.8: Редактирование файла

Создаю и запускаю исполняемый файл lab6-2 (рис. 2.9).



```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
106

```

Рис. 2.9: Запуск исполняемого файла

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 2.10).

```

lab6-2.asm
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit

```

Рис. 2.10: Редактирование файла

Создаю и запускаю новый исполняемый файл , теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.(рис. 2.11).

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10

```

Рис. 2.11: Запуск исполняемого файла

Заменяю в тексте программы функцию iprintLF на iprint (рис. 2.12).

```

lab6-2.asm
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprint
9 call quit

```

Рис. 2.12: Редактирование файла

Создаю и запускаю новый исполняемый файл, вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией `iprintLF`, а `iprint` не добавляет к выводу символ переноса строки, в отличие от `iprintLF` (рис. 2.13).

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 2.13: Запуск исполняемого файла

## 2.1 Выполнение арифметических операций в NASM

Создаю файл `lab7-3.asm` (рис. 2.14).

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ touch lab7-3.asm

```

Рис. 2.14: Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения  $f(x) = (5 * 2 + 3)/3$  (рис. 2.15).

```

lab6-3.asm
1 %include 'in_out.asm' ; подключение внешнего
  файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ---- Вычисление выражения
9 mov eax,5 ; EAX=5
10 mov ebx,2 ; EBX=2
11 mul ebx ; EAX=EAX*EBX
12 add eax,3 ; EAX=EAX+3
13 xor edx,edx ; обнуляем EDX для корректной работы
  div
14 mov ebx,3 ; EBX=3
15 div ebx ; EAX=EAX/3, EDX=остаток от деления
16 mov edi,eax ; запись результата вычисления в
  'edi'
17 ; ---- Вывод результата на экран
18 mov eax,div ; вызов подпрограммы печати
19 call sprint ; сообщения 'Результат: '
20 mov eax,edi ; вызов подпрограммы печати значения
21 call iprintLF ; из 'edi' в виде символов
22 mov eax,rem ; вызов подпрограммы печати
23 call sprint ; сообщения 'Остаток от деления: '
24 mov eax,edx ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edx' (остаток) в виде
  символов
26 call quit ; вызов подпрограммы завершения

```

Рис. 2.15: Редактирование файла

Создаю исполняемый файл и запускаю его (рис. 2.16).

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab6$ nasm -f elf lab6-3.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab6$ ld -m elf_i386 -o lab6-3 lab6-3.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab6$ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 2.16: Запуск исполняемого файла

Изменяю программу так, чтобы она вычисляла значение выражения  $f(x) = (4 * 6 + 2)/5$  (рис. 2.17).

```

*lab6-3.asm
1 %include 'in_out.asm' ; подключение внешнего
  файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ---- Вычисление выражения
9 mov eax,4 ; EAX=4
10 mov ebx,6 ; EBX=6
11 mul ebx ; EAX=EAX*EBX
12 add eax,2 ; EAX=EAX+2
13 xor edx,edx ; обнуляем EDX для корректной работы
  div
14 mov ebx,5 ; EBX=5
15 div ebx ; EAX=EAX/5, EDX=остаток от деления
16 mov edi,eax ; запись результата вычисления в
  'edi'
17 ; ---- Вывод результата на экран
18 mov eax,div ; вызов подпрограммы печати
19 call sprint ; сообщения 'Результат: '
20 mov eax,edi ; вызов подпрограммы печати значения
21 call iprintLF ; из 'edi' в виде символов
22 mov eax,rem ; вызов подпрограммы печати
23 call sprint ; сообщения 'Остаток от деления: '
24 mov eax,edx ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edx' (остаток) в виде
  СИМВОЛОВ
26 call quit ; вызов подпрограммы завершения

```

Рис. 2.17: Изменение программы

Создаю и запускаю новый исполняемый файл (рис. 2.18).

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

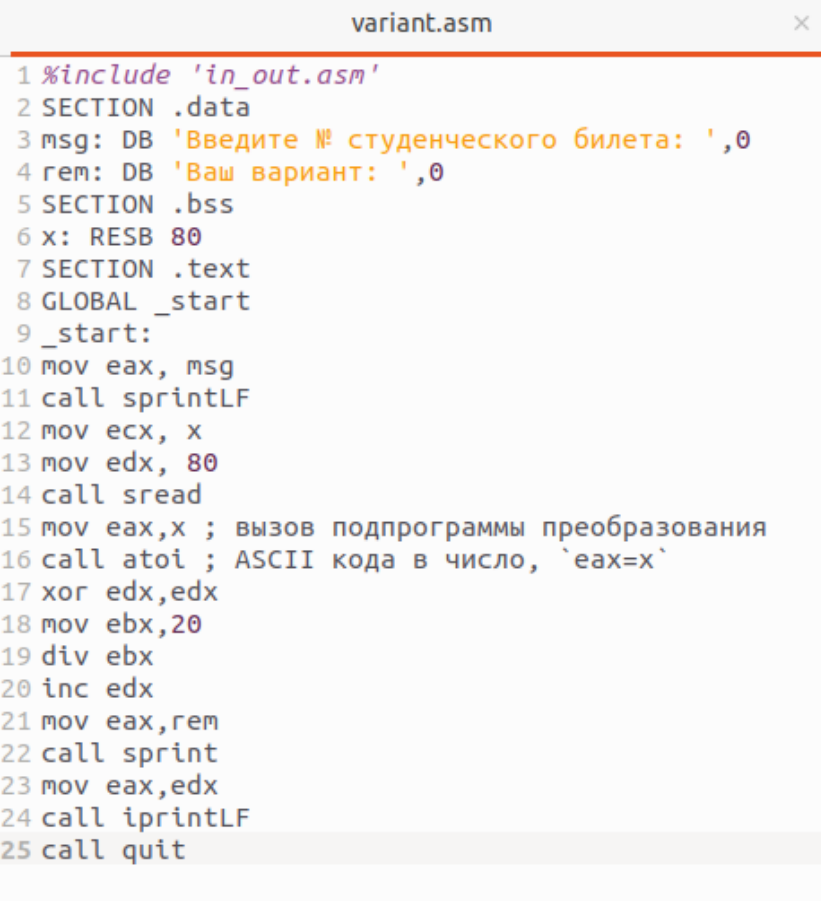
Рис. 2.18: Запуск исполняемого файла

Создаю файл variant.asm (рис. 2.19).

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ touch variant.asm
```

Рис. 2.19: Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 2.20).



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите № студенческого билета: ',0
4 rem: DB 'Ваш вариант: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintLF
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax,x ; вызов подпрограммы преобразования
16 call atoi ; ASCII кода в число, `eax=x`
17 xor edx,edx
18 mov ebx,20
19 div ebx
20 inc edx
21 mov eax,rem
22 call sprint
23 mov eax,edx
24 call iprintLF
25 call quit
```

Рис. 2.20: Редактирование файла

Создаю и запускаю исполняемый файл , ввожу номер своего студ. билета, мой вариант -6 (рис. 2.21).

```

avsideljnikov@avsideljnikov-VirtualBox: ~/work/arch-pc/lab06$ nasm -f elf variant.asm
avsideljnikov@avsideljnikov-VirtualBox: ~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
avsideljnikov@avsideljnikov-VirtualBox: ~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132236085
Ваш вариант: 6

```

Рис. 2.21: Запуск исполняемого файла

## 2.2 Ответы на вопросы

1. За вывод сообщения “Ваш вариант” отвечают строки кода: `mov eax,rem`  
`call sprint`
2. Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки  
`call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`
4. За вычисления варианта отвечают строки: `xor edx,edx` ; обнуление `edx` для корректной работы `div` `mov ebx,20` ; `ebx = 20` `div ebx` ; `eax = eax/20`, `edx` - остаток от деления `inc edx` ; `edx = edx + 1`
5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1
7. За вывод на экран результатов вычислений отвечают строки: `mov eax,edx`  
`call iprintLF`

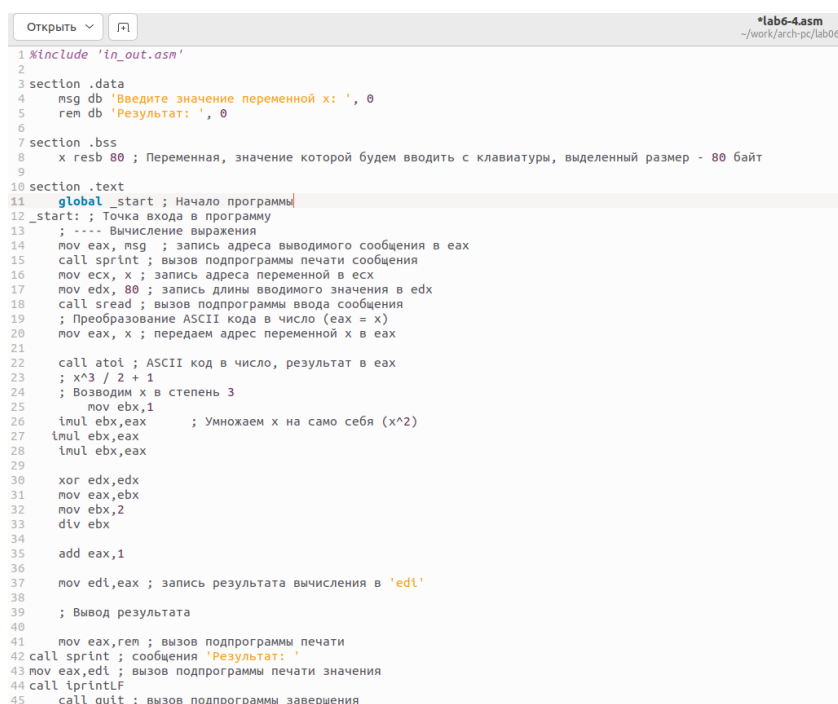
## 2.3 Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm (рис. 2.22).

```
avsideIjnikov@avsideIjnikov-VirtualBox:~/work/arch-pc/lab06$ touch lab6-4.asm
```

Рис. 2.22: Создание файла

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения  $x^3 / 2 + 1$  (Вариант 6) (рис. 2.23).



```
1 %include 'in_out.asm'
2
3 section .data
4     msg db 'Введите значение переменной x: ', 0
5     res db 'Результат: ', 0
6
7 section .bss
8     x resb 80 ; Переменная, значение которой будем вводить с клавиатуры, выделенный размер - 80 байт
9
10 section .text
11     global _start ; Начало программы
12 _start: ; Точка входа в программу
13     ; --- Вычисление выражения
14     mov eax, msg ; запись адреса выводимого сообщения в eax
15     call sprint ; вызов подпрограммы печати сообщения
16     mov ecx, x ; запись адреса переменной в ecx
17     mov edx, 80 ; запись длины вводимого значения в edx
18     call sread ; вызов подпрограммы ввода сообщения
19     ; Преобразование ASCII кода в число (eax = x)
20     mov eax, x ; передаем адрес переменной x в eax
21
22     call atoi ; ASCII код в число, результат в eax
23     ; x^3 / 2 + 1
24     ; Возводим x в степень 3
25     mov ebx, 1
26     imul ebx, eax ; Умножаем x на само себя (x^2)
27     imul ebx, eax
28     imul ebx, eax
29
30     xor edx, edx
31     mov eax, ebx
32     mov ebx, 2
33     div ebx
34
35     add eax, 1
36
37     mov edi, eax ; запись результата вычисления в 'edi'
38
39     ; Вывод результата
40
41     mov eax, res ; вызов подпрограммы печати
42     call sprint ; сообщения 'Результат: '
43     mov eax, edi ; вызов подпрограммы печати значения
44     call iprintf
45     call quit ; вызов подпрограммы завершения
```

Рис. 2.23: Написание программы

Создаю и запускаю исполняемый файл, при вводе значений 1 и 5 выдаёт правильные значения 5 и 63 (рис. 2.24)

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-4
Введите значение переменной x: 2
Результат: 5
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-4
Введите значение переменной x: 5
Результат: 63
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 2.24: Запуск исполняемого файла

```
“%include ‘in_out.asm’
```

```
section .data msg db ‘Введите значение переменной x:’, 0 rem db ‘Результат:’, 0
```

```
section .bss x resb 80 ; Переменная, значение которой будем вводить с клавиатуры, выделенный размер - 80 байт
```

```
section .text global _start ; Начало программы _start: ; Точка входа в программу ;
```

```
— Вычисление выражения mov eax, msg ; запись адреса выводимого сообщения в eax
```

```
call sprint ; вызов подпрограммы печати сообщения
```

```
mov ecx, x ; запись адреса переменной в ecx
```

```
mov edx, 80 ; запись длины вводимого значения в edx
```

```
call sread ; вызов подпрограммы ввода сообщения
```

```
; Преобразование ASCII кода в число (eax = x)
```

```
mov eax, x ; передаем адрес переменной x в eax
```

```
call atoi ; ASCII код в число, результат в eax
```

```
;  $x^3 / 2 + 1$ 
```

```
; Возводим x в степень 3
```

```
mov ebx, 1
```



```
imul ebx,eax ; Умножаем x на само себя (x^2)
```

```
imul ebx,eax
```

```
imul ebx,eax
```

```
xor edx,edx
```

```
mov eax,ebx
```

```
mov ebx,2
```

```
div ebx
```

```
add eax,1
```

```
mov edi,eax ; запись результата вычисления в 'edi'
```

```
; Вывод результата
```

```
mov eax,tem ; вызов подпрограммы печати
```

```
call sprint ; сообщения 'Результат:'  
mov eax,edi ; вызов подпрограммы печати  
значения  
call iprintLF  
call quit ; вызов подпрограммы завершения“
```

## **3 Выводы**

При выполнении данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.