

Лабораторная работа №6

Арифметические операции в NASM.

Славинский Владислав Вадимович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение задания для самостоятельной работы	18
4	Вывод	21

Список иллюстраций

2.1	Создание lab6	6
2.2	Программа	7
2.3	Запуск	7
2.4	Замена	8
2.5	Запуск	8
2.6	Создание файла lab6-2.asm	9
2.7	Программа	9
2.8	Запуск	9
2.9	Программа	10
2.10	Запуск	10
2.11	Замена	11
2.12	Запуск с заменой	11
2.13	Создание lab6-3.asm	11
2.14	Вычисление выражения	12
2.15	Запуск вычисления выражения	13
2.16	Новое выражение	14
2.17	Запуск	15
2.18	Создание файла	15
2.19	Код вычисления варианта задания по номеру	16
2.20	Запуск	17
3.1	Создание laba6.asm	18
3.2	Код программы для вычисления уравнение	19
3.3	Запуск программы	20

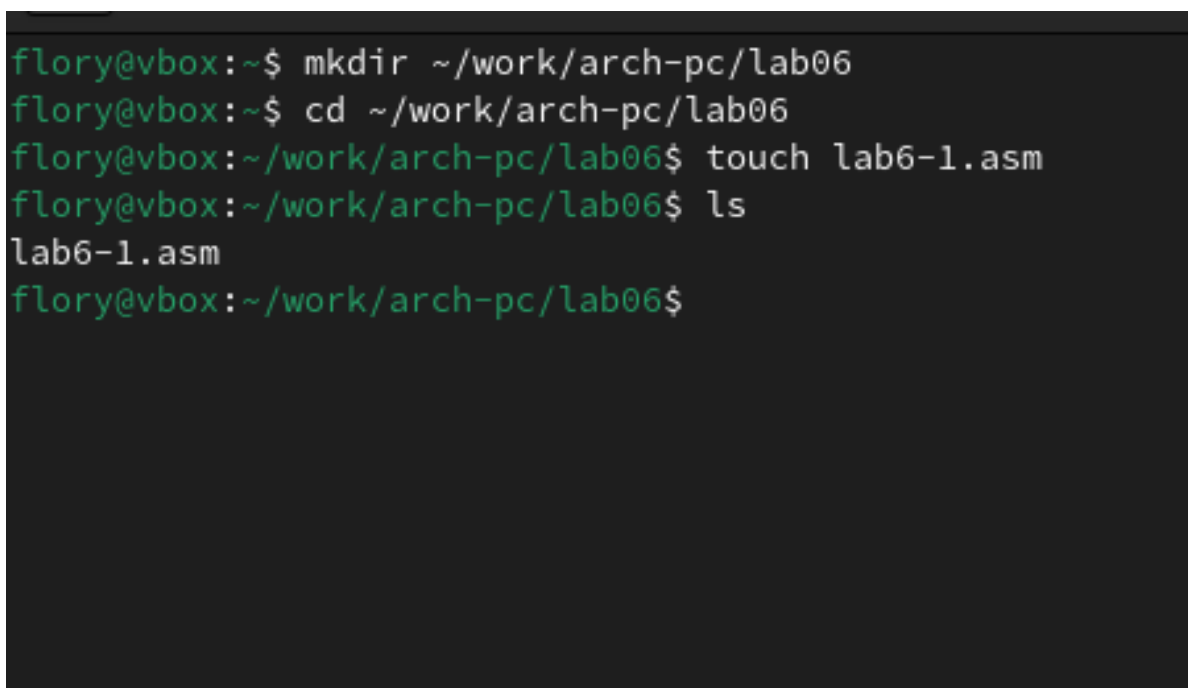
Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

1) Создал каталог для лабораторной работы №6 и создал файл lab6-1.asm: (Рис. 2.1)

A screenshot of a terminal window with a dark background and green text. The terminal shows a series of commands and their outputs. The user 'flory' is at a machine named 'vbox'. The commands are: 'mkdir ~/work/arch-pc/lab06', 'cd ~/work/arch-pc/lab06', 'touch lab6-1.asm', and 'ls'. The output of 'ls' is 'lab6-1.asm'.

```
flory@vbox:~$ mkdir ~/work/arch-pc/lab06
flory@vbox:~$ cd ~/work/arch-pc/lab06
flory@vbox:~/work/arch-pc/lab06$ touch lab6-1.asm
flory@vbox:~/work/arch-pc/lab06$ ls
lab6-1.asm
flory@vbox:~/work/arch-pc/lab06$
```

Рис. 2.1: Создание lab6

2) Ввел код программы вывода значения регистра eax: (Рис. 2.2)

```

lab6-1.asm      [----]  0 L:[ 1+15 16/ 17] *(166 / 176b
#include 'in_out.asm'

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov  eax,'6'
mov  ebx,'4'
add  eax,ebx
mov  [buf1],eax
mov  eax,buf1
call sprintfLF
call quit

```

Рис. 2.2: Программа

3)Создал исполняемый файл и запустил его: (Рис. 2.3)

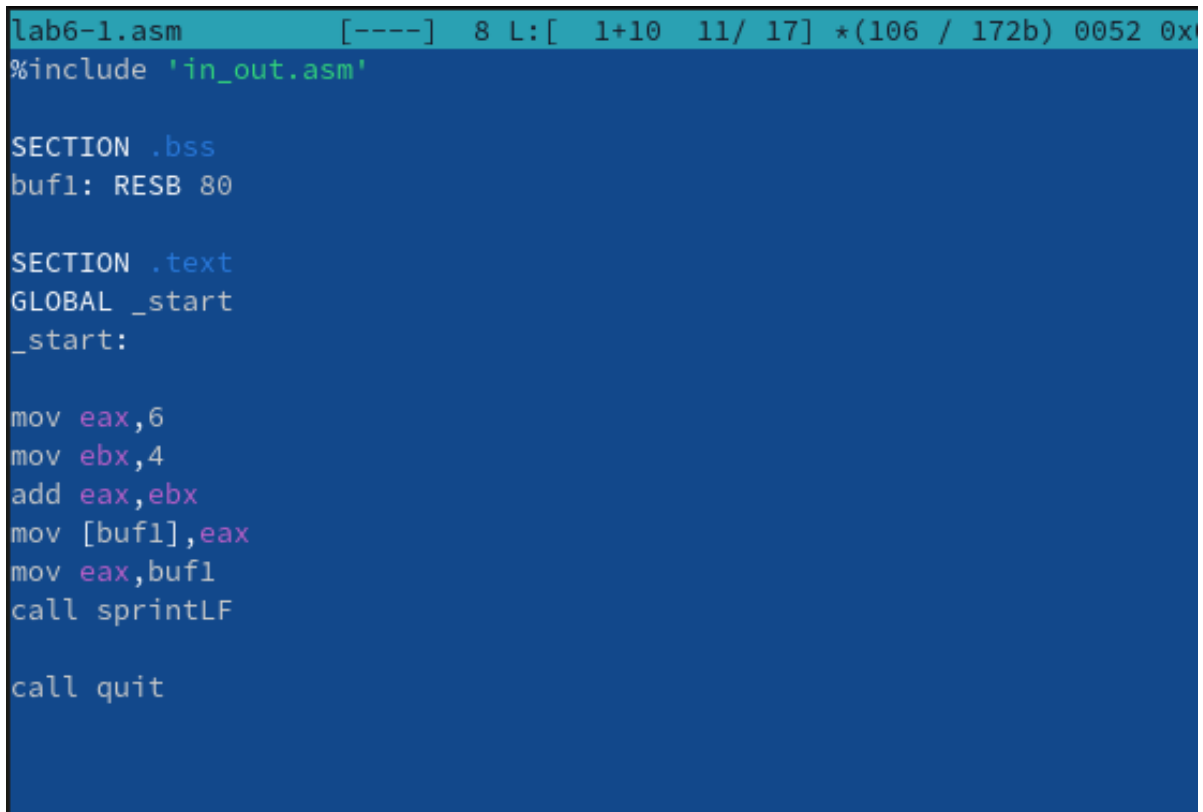
```

flory@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
flory@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
flory@vbox:~/work/arch-pc/lab06$ ./lab6-1
j
flory@vbox:~/work/arch-pc/lab06$

```

Рис. 2.3: Запуск

4)Заменял символы на числа: (Рис. 2.4)



```
lab6-1.asm      [----]  8 L:[  1+10  11/ 17] *(106 / 172b) 0052 0x
#include 'in_out.asm'

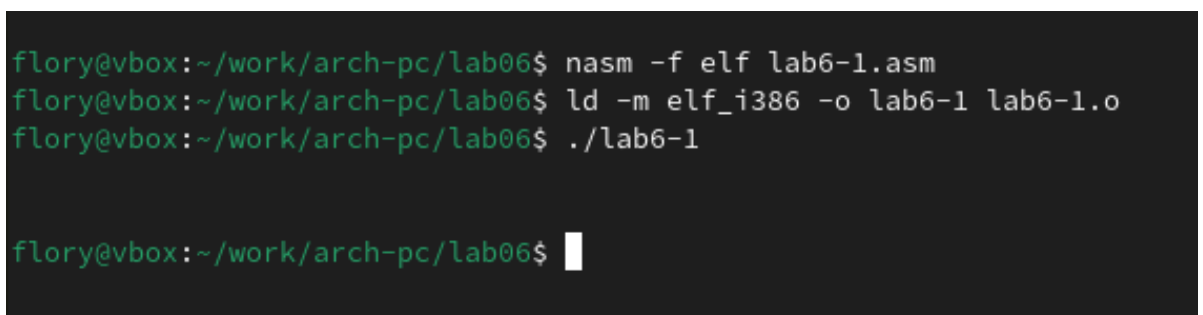
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov  eax,6
mov  ebx,4
add  eax,ebx
mov  [buf1],eax
mov  eax,buf1
call sprintf
call quit
```

Рис. 2.4: Замена

5)Запустил измененный файл(Рис. 2.5)



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

flory@vbox:~/work/arch-pc/lab06$
```

Рис. 2.5: Запуск

6)Создал файл lab6-2.asm (Рис. 2.6)


```
flory@vbox:~/work/arch-pc/lab06$ touch lab6-2.asm
flory@vbox:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2.asm
flory@vbox:~/work/arch-pc/lab06$
```

Рис. 2.6: Создание файла lab6-2.asm

7) Ввел также код программы вывода значения регистра eax: (Рис. 2.7)

```
lab6-2.asm  [-M--]  0 L:[ 1+10 11/ 12] *(110 / 120b) 001
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov  eax, '6'
mov  ebx, '4'
add  eax, ebx
call iprintLF

call quit
```

Рис. 2.7: Программа

8) Запустил его. (Рис. 2.8)

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

Рис. 2.8: Запуск

9) Заменяю символы на числа. (Рис. 2.9)

```

lab6-2.asm      [----]  9 L:[  1+ 8   9/ 12] *(89 / 116b) 0098 0x062
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov  eax,6
mov  ebx,4
add  eax,ebx
call iprintLF

call quit

```

Рис. 2.9: Программа

10)Запустил его:(Рис. 2.10)

```

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

```

Рис. 2.10: Запуск

11)Заменил iprintLF на iprint:(Рис. 2.11)

```
lab6-2.asm [----] 11 L:[ 1+ 9 10/ 12] *(103 / 114b) 0010 0x00
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit
```

Рис. 2.11: Замена

- 12) Запустил исполняемый файл и заметил, что iprint не переносит строку:(Рис. 2.12)

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

Рис. 2.12: Запуск с заменой

- 13)Создал файл lab6-3.asm:(Рис. 2.13)

```
flory@vbox:~/work/arch-pc/lab06$ touch lab6-3.asm
flory@vbox:~/work/arch-pc/lab06$ ls
in_out.asm lab6-1.asm lab6-2 lab6-2.o
lab6-1 lab6-1.o lab6-2.asm lab6-3.asm
flory@vbox:~/work/arch-pc/lab06$
```

Рис. 2.13: Создание lab6-3.asm

- 14)Записал код программы вычисления выражения:(Рис. 2.14)

```

lab6-3.asm      [----]  0 L:[ 1+37 38/ 39] *(1306/1375b) 0010 0x00A
;-----
; Программа вычисления выражения
;-----

%include 'in_out.asm' ; подключение внешнего файла

SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления

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

; ---- Вывод результата на экран

mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов

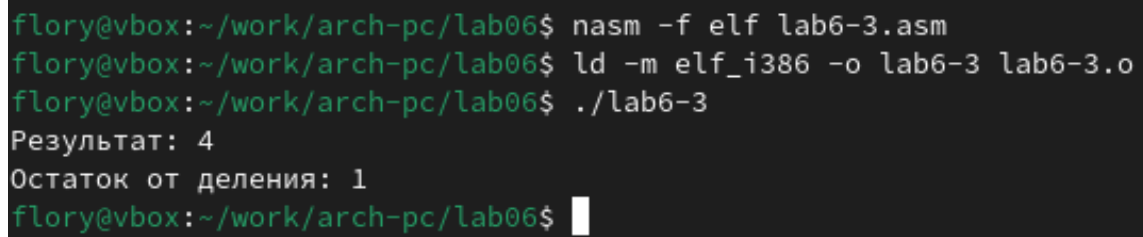
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов

call quit ; вызов подпрограммы завершения

```

Рис. 2.14: Вычисление выражения

15)Создал исполняемый файл и запустил его:(Рис. 2.15)



```
flory@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
flory@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
flory@vbox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
flory@vbox:~/work/arch-pc/lab06$
```

Рис. 2.15: Запуск вычисления выражения

16)Исправил код для вычисления нового выражения:(Рис. 2.16)

```

lab6-3.asm      [----] 19 L:[ 1+22 23/ 39] *(585 /1375b) 0044 0x02C
;-----
; Программа вычисления выражения
;-----

%include 'in_out.asm' ; подключение внешнего файла

SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

; ---- Вычисление выражения
mov eax,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления

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

; ---- Вывод результата на экран

mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов

mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов

call quit ; вызов подпрограммы завершения

```

Рис. 2.16: Новое выражение

17)Создал исполняемый файл и запустил:(Рис. 2.17)

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

Рис. 2.17: Запуск

18) Теперь перейду к вычислению варианта задания по номеру студ. билета. Для этого создам файл variant.asm :(Рис. 2.18)

```
flory@vbox:~/work/arch-pc/lab06$ touch variant.asm
flory@vbox:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2  lab6-2.asm  lab6-2.o  lab6-3  lab6-3.asm  lab6-3.o  variant.asm
flory@vbox:~/work/arch-pc/lab06$
```

Рис. 2.18: Создание файла

19) Введу код программы вычисления варианта задания по номеру студ. билета(Рис. 2.19)

```

variant.asm      [----]  0 L:[ 1+23 24/ 38] *(399 / 628b) 0010 0x00A
;-----
; Программа вычисления варианта
;-----

%include 'in_out.asm'

SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
call sread

mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x`

xor edx, edx
mov ebx, 20
div ebx
inc edx

mov eax, rem
call sprint
mov eax, edx
call iprintLF

call quit

```

Рис. 2.19: Код вычисления варианта задания по номеру

20)Создам исполняемый файл и запущу его:(Рис. 2.20)

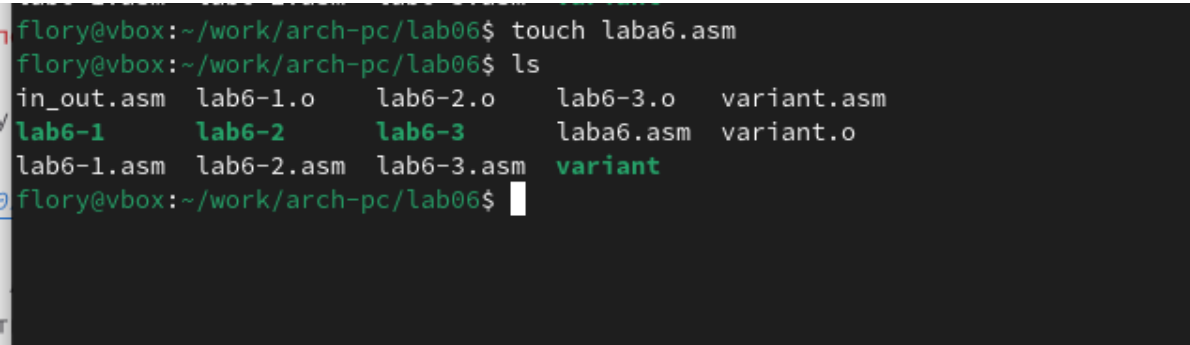

```
flory@vbox:~/work/arch-pc/lab06$ nasm -f elf variant.asm
flory@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
flory@vbox:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246169
Ваш вариант: 10
flory@vbox:~/work/arch-pc/lab06$
```

Рис. 2.20: Запуск

Ответы на вопросы: 1) Какие строки листинга 6.4 отвечают за вывод на экран сообщения “Ваш вариант:”? - `move eax,rem call sprint`. 2) Для чего используется следующие инструкции?`mov ecx, x``mov edx,80 call sread` - эти конструкции обеспечивают ввод строки от пользователя. 3) Для чего используется инструкция “`call atoi`”? - помогает программе работать с числовыми значениями, в виде целых чисел, а не строк. 4) Какие строки листинга 6.4 отвечают за вычисления варианта? - `xor edx, edx``move ebx,20 div ebx inc edx` 5) В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”? - `edx` 6) Для чего используется инструкция “`inc edx`”? - Для увеличения `edx` на 1. 7) Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений? - `move eax, rem call sprint`
`move eax, edx call iprintLF`

3 Выполнение задания для самостоятельной работы

1) Создал файл laba6.asm для выполнения самостоятельного задания и заполнил его (Рис. 3.1)



```
flory@vbox:~/work/arch-pc/lab06$ touch laba6.asm
flory@vbox:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.o    lab6-2.o    lab6-3.o    variant.asm
lab6-1      lab6-2     lab6-3      laba6.asm   variant.o
lab6-1.asm  lab6-2.asm lab6-3.asm  variant
flory@vbox:~/work/arch-pc/lab06$
```

Рис. 3.1: Создание laba6.asm

2) Написал код программы для вычисления уравнения $5(x+18)-28$ (Рис. 3.2)

```

[lab6.asm      [----]  6 L:[  1+19  20/ 36] *(238 / 397b) 0114 0x072
%include 'in_out.asm'

SECTION .data
msg:DB 'Введите значение X',0
final:DB 'Ответ:',0

SECTION .bss
x: RESB 80
rez: RESB 80

SECTION .start
GLOBAL _start
_start:

mov eax,msg
call sprintLF

mov ecx,x
mov edx,80
call sread
mov eax,x
call atoi

add eax,18
mov ebx,5
mul ebx
xor ebx,ebx
add eax, -28
mov [rez],eax

mov eax,final
call sprint
mov eax,[rez]
call iprintLF

call quit

```

Рис. 3.2: Код программы для вычисления уравнение

3) Создал исполняемый файл и запустил его, введя значения для $x_1=2$, $x_2=3$ (Рис. 3.3)

```
flory@vbox:~/work/arch-pc/lab06$ nasm -f elf laba6.asm
flory@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o laba6 laba6.o
ld: невозможно найти laba6.o: Нет такого файла или каталога
flory@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o laba6 laba6.o
flory@vbox:~/work/arch-pc/lab06$ ./laba6
Введите значение X
2
Ответ:72
flory@vbox:~/work/arch-pc/lab06$ ./laba6
Введите значение X
3
Ответ:77
flory@vbox:~/work/arch-pc/lab06$
```

Рис. 3.3: Запуск программы

4 Вывод

В ходе выполнения лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.