

# **Лабораторная работа №7**

**Арифметические операции в NASM**

Попова Елизавета Сергеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Вопросы</b>	<b>15</b>
<b>4</b>	<b>Самостоятельная работа</b>	<b>16</b>
<b>5</b>	<b>Выводы</b>	<b>19</b>

# Список иллюстраций

2.1	Создание файла lab7-1.asm . . . . .	6
2.2	Текст в файле lab7-1.asm . . . . .	7
2.3	Создание файла и проверка работы . . . . .	7
2.4	Изменения программы lab7-1.asm . . . . .	8
2.5	Проверка программы . . . . .	8
2.6	Создание файла lab7-2.asm . . . . .	8
2.7	lab7-2.asm . . . . .	9
2.8	Запуск программы lab7-2.asm . . . . .	9
2.9	Проверка программы . . . . .	9
2.10	Замена iprintLF на iprint . . . . .	10
2.11	Запуск программы с изменениями . . . . .	10
2.12	Текст в программе lab7-3.asm . . . . .	11
2.13	Запуск программы lab7-3 . . . . .	11
2.14	Изменения в программе lab7-3 . . . . .	12
2.15	Запуск программы . . . . .	12
2.16	Текст программы в файле variant.asm . . . . .	13
2.17	Запуск программы variant . . . . .	14
4.1	Программа для вычисления записанная в файле 1.asm . . . . .	17
4.2	Результаты работы этой программы . . . . .	18

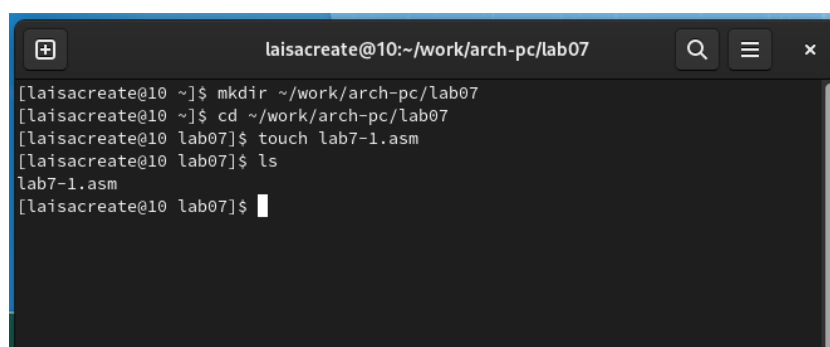
## List of Tables

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Выполнение лабораторной работы

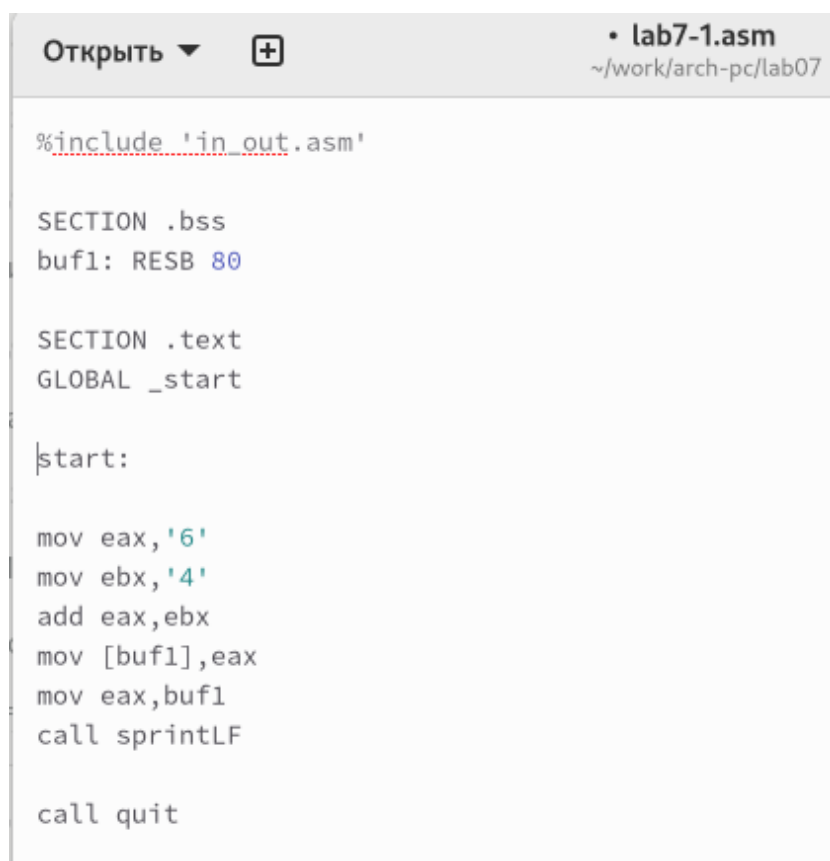
- 1) Я создала каталог lab07 и внутри создал файл lab7-1.asm.



```
laisacreate@10:~/work/arch-pc/lab07
[laissacreate@10 ~]$ mkdir ~/work/arch-pc/lab07
[laissacreate@10 ~]$ cd ~/work/arch-pc/lab07
[laissacreate@10 lab07]$ touch lab7-1.asm
[laissacreate@10 lab07]$ ls
lab7-1.asm
[laissacreate@10 lab07]$
```

Рис. 2.1: Создание файла lab7-1.asm

- 2) В файл lab7-1.asm я ввела нужный текст, создала файл и проверила его.  
Программа вывела j.



```
Открыть ▾ + • lab7-1.asm
~/.work/arch-pc/lab07

%include 'in_out.asm'

SECTION .bss
buf1: RESB 80

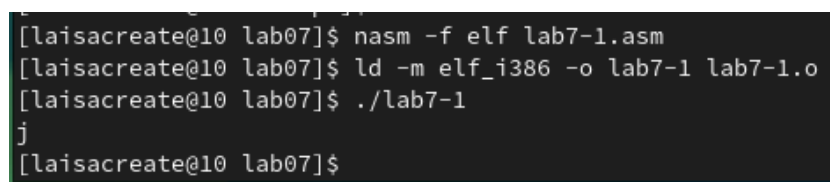
SECTION .text
GLOBAL _start

_start:

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

call quit
```

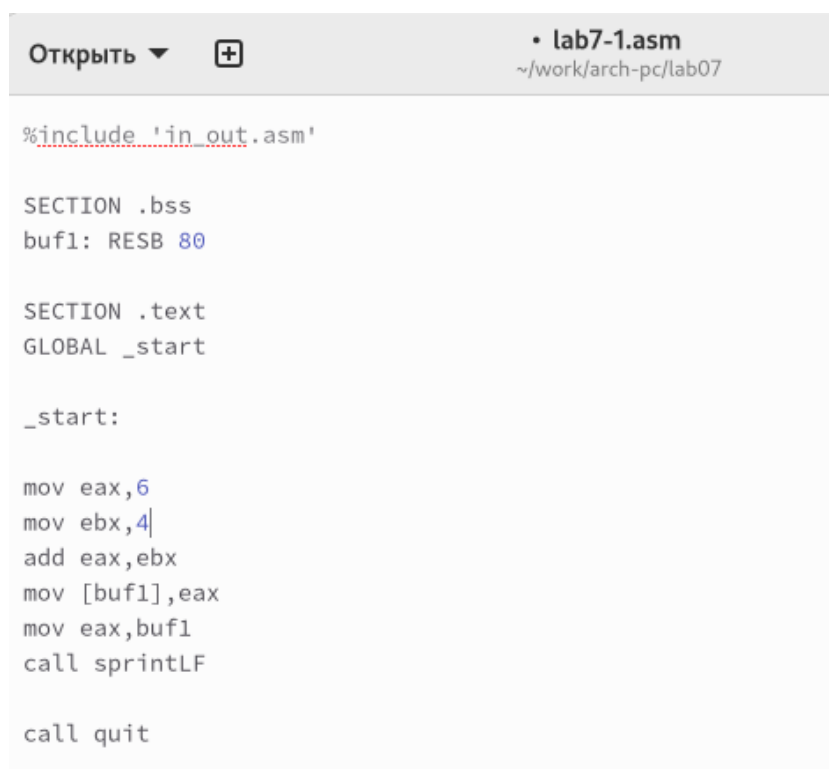
Рис. 2.2: Текст в файле lab7-1.asm



```
[laisacrete@10 lab07]$ nasm -f elf lab7-1.asm
[laisacrete@10 lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[laisacrete@10 lab07]$ ./lab7-1
j
[laisacrete@10 lab07]$
```

Рис. 2.3: Создание файла и проверка работы

- 3) Я изменила текст программы. Когда я сделала программу и запустила ее, она вывела невидимый символ, как и должна была.



```
Открыть ▾ + • lab7-1.asm
~/.work/arch-pc/lab07

%include 'in_out.asm'

SECTION .bss
buf1: RESB 80

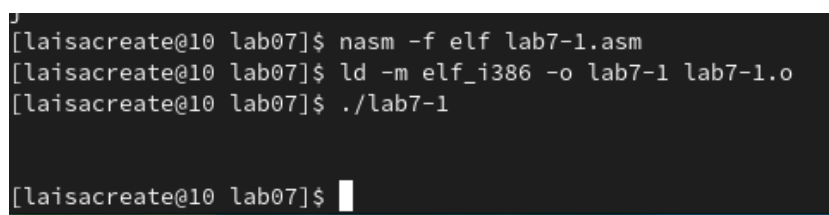
SECTION .text
GLOBAL _start

_start:

mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF

call quit
```

Рис. 2.4: Изменения программы lab7-1.asm

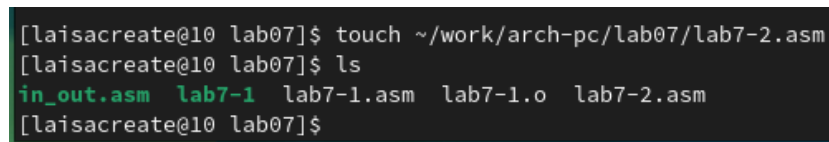


```
[laisacrete@10 lab07]$ nasm -f elf lab7-1.asm
[laisacrete@10 lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[laisacrete@10 lab07]$ ./lab7-1

[laisacrete@10 lab07]$
```

Рис. 2.5: Проверка программы

4) Я создала файл lab7-2.asm и ввел в него текст программы.



```
[laisacrete@10 lab07]$ touch ~/.work/arch-pc/lab07/lab7-2.asm
[laisacrete@10 lab07]$ ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o lab7-2.asm
[laisacrete@10 lab07]$
```

Рис. 2.6: Создание файла lab7-2.asm





```
Открыть + • lab7-2.asm
~/.work/arch-pc/lab07

%include 'in_out.asm'

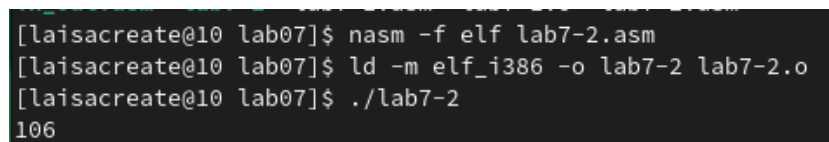
SECTION .text
GLOBAL _start
_start:

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

call quit|
```

Рис. 2.7: lab7-2.asm

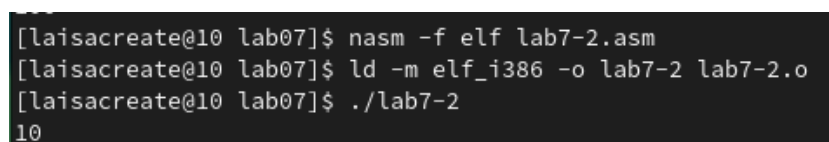
- 5) Создала исполняемый файл и запустил его. Как и предполагалось, он вывел число 106.



```
[laisacrete@10 lab07]$ nasm -f elf lab7-2.asm
[laisacrete@10 lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[laisacrete@10 lab07]$ ./lab7-2
106
```

Рис. 2.8: Запуск программы lab7-2.asm

- 6) Я изменила текст программы lab7-2 и запустила ее. Она вывела число 10, как и должна была.



```
[laisacrete@10 lab07]$ nasm -f elf lab7-2.asm
[laisacrete@10 lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[laisacrete@10 lab07]$ ./lab7-2
10
```

Рис. 2.9: Проверка программы

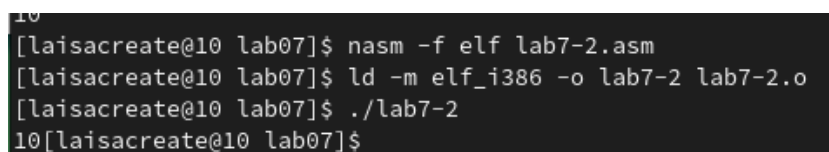
- 7) Я заменила функцию iprintLF на iprint. Проверила файл и результат отличался в выводе данных. Когда используется команда iprintLF, результат

выводится на следующей строке, а при использовании команды `iprint` на этой же строке.



```
Открыть ▾ + • lab7-2.asm  
~/work/arch-pc/lab07  
  
%include 'in_out.asm'  
  
SECTION .text  
GLOBAL _start  
_start:  
  
mov eax,6  
mov ebx,4  
  
add eax,ebx  
call iprint  
call quit
```

Рис. 2.10: Замена `iprintLF` на `iprint`



```
10  
[laisacrete@10 lab07]$ nasm -f elf lab7-2.asm  
[laisacrete@10 lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o  
[laisacrete@10 lab07]$ ./lab7-2  
10[laisacrete@10 lab07]$
```

Рис. 2.11: Запуск программы с изменениями

- 8) Я создала файл `lab7-3.asm` и ввел текст программы. Запустила и проверила ее работу.

```

Открыть ▾ + • lab7-3.asm
~/work/arch-pc/lab07

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 iprintfLF ; 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 iprintfLF ; из 'edx' (остаток) в виде символов

call quit ; вызов подпрограммы завершения

```

Рис. 2.12: Текст в программе lab7-3.asm

```

[laisacrete@10 lab07]$ nasm -f elf lab7-3.asm
[laisacrete@10 lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[laisacrete@10 lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[laisacrete@10 lab07]$

```

Рис. 2.13: Запуск программы lab7-3

- 9) Я изменила программу для примера  $F(x) = (4*6+2)/5$ . Запустила ее и проверила.

```

;-----
; 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=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX

add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; 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: Изменения в программе lab7-3

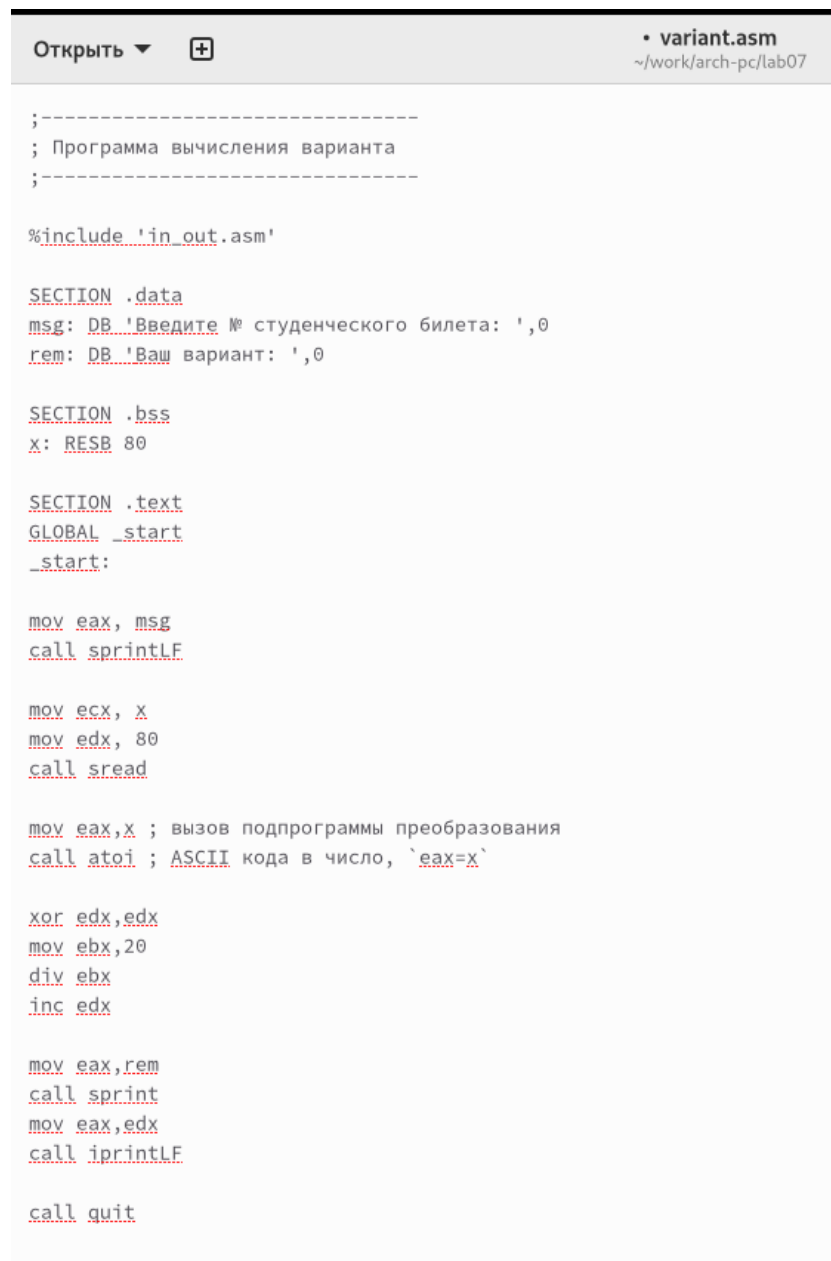
```

[laisacrete@10 lab07]$ nasm -f elf lab7-3.asm
[laisacrete@10 lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[laisacrete@10 lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[laisacrete@10 lab07]$

```

Рис. 2.15: Запуск программы

10) Я создала файл variant.asm и ввела туда нужный текст.



```
Открыть ▾ + • variant.asm
~/work/arch-pc/lab07

;-----
; 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 sprintf

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 sprintf
mov eax, edx
call iprintfLF

call quit
```

Рис. 2.16: Текст программы в файле variant.asm

11) Я ввела в эту программу свой студенческий билет. Он вывел ответ 15. Я проверила это аналитически, ответ совпал.

```
[laisacrete@10 lab07]$ nasm -f elf variant.asm
[laisacrete@10 lab07]$ ld -m elf_i386 -o variant variant.o
[laisacrete@10 lab07]$ ./variant
Введите № студенческого билета:
1132226514
Ваш вариант: 15
[laisacrete@10 lab07]$
```

Рис. 2.17: Запуск программы variant

### 3 Вопросы

- 1) Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? Ответ: `mov eax,msg call sprintLF`
- 2) Для чего используются следующие инструкции? `asm mov ecx, x mov edx, 80 call sread`. Ответ: Эти инструкции используются для ввода переменной X с клавиатуры и сохранения введенных данных.
- 3) Для чего используется инструкция “`call atoi`”? Ответ: Эта инструкция используется для преобразования кода переменной ASCII в число.
- 4) Какие строки листинга 7.4 отвечают за вычисления варианта? Ответ: `mov ebx,20 div ebx inc edx`
- 5) В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”? Ответ: В регистре `ebx`.
- 6) Для чего используется инструкция “`inc edx`”? Ответ: Для увеличения значения `edx` на 1.
- 7) Какие строки листинга 7.4 отвечают за вывод на экран результата вычисления? Ответ: `mov eax,edx call iprintLF`

## 4 Самостоятельная работа

Я написала программу которая будет решать выражение и выводить ответ при введенных  $x$ . Так как в 11 пункте мне выдало 15 вариантов, то программу я писала для выражения  $F(x) = (5+x)^2-3$ . Я вводила значения переменных 5 и 1, ответы верны.



```

%include 'in_out.asm'
SECTION .data

div: DB '(5+x)^2 - 3',0
msg: DB 'Введите значение x: ',0
rem: DB 'Ответ при x = ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,div
call sprintLF

mov eax,msg
call sprintLF

mov ecx,x
mov edx,80
call sread

mov eax,x
call atoi

xor edx,edx

add eax,5
xor ebx,ebx
mul eax
sub eax,3

mov edi,eax

mov eax,rem
call sprintLF
mov eax,edi
call iprintLF

call quit

```

Рис. 4.1: Программа для вычисления записанная в файле 1.asm

```
[laisacrete@10 lab07]$ nasm -f elf 1.asm
[laisacrete@10 lab07]$ ld -m elf_i386 -o 1 1.o
[laisacrete@10 lab07]$ ./1
(5+x)^2 - 3
Введите значение x:
5
Ответ при x =
97
[laisacrete@10 lab07]$ ./1
(5+x)^2 - 3
Введите значение x:
1
Ответ при x =
33
[laisacrete@10 lab07]$
```

Рис. 4.2: Результаты работы этой программы

## 5 Выводы

Я освоила арифметические инструкции языка ассемблер NASM.