

# **Лабораторная работа №6**

**Арифметические операции в NASM.**

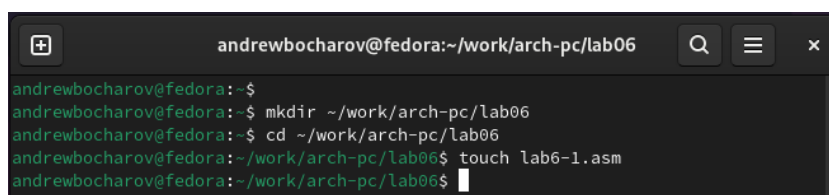
Бочаров Андрей

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>3</b>
<b>2</b>	<b>Выполнение самостоятельной работы</b>	<b>15</b>
<b>3</b>	<b>Выводы</b>	<b>18</b>

# 1 Цель работы

Целью работы является изучение арифметические инструкции языка ассемблера NASM # Выполнение лабораторной работы создал и першел в каталог для 6 лабораторной работы и командой touch сделал файл lab6-1.asm (рис. 1.1).



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

Рис. 1.1: Каталог lab06

Переписал код из листинга 6.1 (рис. 1.2).



```
GNU nano 7.2 lab6-1.asm
#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. 1.2: Листинг кода

Листинг кода 6.1:

```
%include 'in_out.asm'
```

```

SECTION .bss
buf1: RESB 80

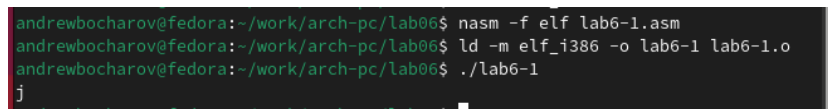
SECTION .text
GLOBAL _start
_start:

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

call quit

```

Создал исполняемый файл и запустил его. (рис. 1.3).



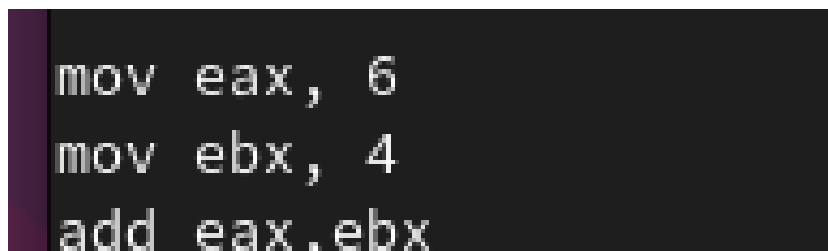
```

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

```

Рис. 1.3: Работает

Заменяю строки с числами в кавычках, на числа без кавычек (рис. 1.4).



```

mov eax, 6
mov ebx, 4
add eax,ebx

```

Рис. 1.4: Обновленный код

Создал исполняемый файл и запустил его. На экран выведен символ переноса

строки, он соответствует коду 10 кодировки ASCII (рис. 1.5).

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

Рис. 1.5: Результат работы файла

Создал файл lab6-1.asm и переписал в него листинг кода 6.2 (рис. 1.6).



```
GNU nano 7.2 lab6-2.asm
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 1.6: Листинг кода

Листинг кода 6.2:

```
%include 'in_out.asm'
```

```
SECTION .text
```

```
GLOBAL _start
```

```
_start:
```

```
mov eax, 6
```

```
mov ebx, 4
```

```
add eax,ebx
```

```
call iprintLF
```

```
call quit
```

Создал исполняемый файл и запустил его. (рис. 1.7).

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

Рис. 1.7: Работает

Вместо символов, записал в регистры числа. Создал исполняемый файл и запустил его. (рис. 1.8).



```
GNU nano 7.2 lab6-2.asm
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 1.8: Обновленные строки кода

Создал исполняемый файл и запустил его. Результат работы число 10.(рис. 1.9).

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

Рис. 1.9: Результат работы

Заменил команду iprintLF на iprint. (рис. 1.10).



andrewbochar

GNU nano 7.2

```
include 'in_out.asm
```

```
SECTION .text
```

```
GLOBAL _start
```

```
_start:
```

```
mov eax, 6
```

```
mov ebx, 4
```

```
add eax, ebx
```

```
call iprint
```

```
call quit
```

Рис. 1.10: Обновленные строки кода

Запустил исполняемый файл, результат вывелся без переноса строки после себя. (рис. 1.11).

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

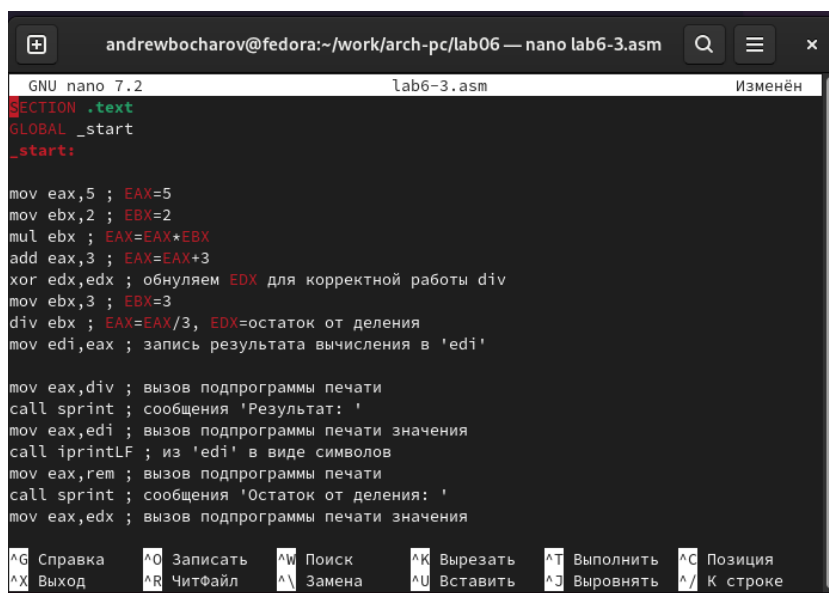
Рис. 1.11: Результат работы

Командой touch сделал файл lab6-3.asm (рис. 1.12).

```
andrewbocharov@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
```

Рис. 1.12: Создание файла

Переписал код из листинга 6.3 (рис. 1.13).



```
GNU nano 7.2 lab6-3.asm
SECTION .text
GLOBAL _start
_start:

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'

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

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция
^X Выход        ^R ЧитФайл     ^\ Замена       ^U Вставить     ^J Выводить     ^/_ К строке
```

Рис. 1.13: Копирование файла

Листинг кода 6.3:

```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
```



```

SECTION .text
GLOBAL _start
_start:

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'

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

```

Создал исполняемый файл и запустил его. Результат совпадает с ожидаемым.  
(рис. 1.14).

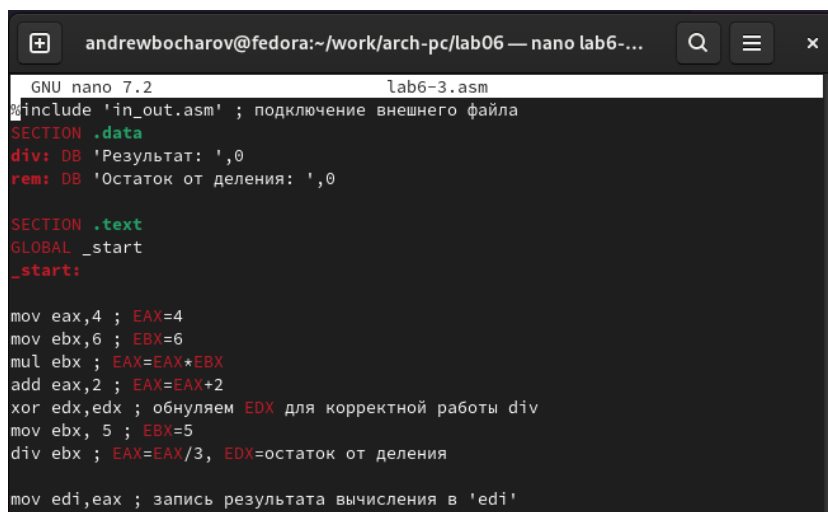
```

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

```

Рис. 1.14: Код работает

Переписал код в файле lab6-3.asm для вычисления выражения  $f(x) = (4 \cdot 6 + 2)/5$ . (рис. 1.15).



```

GNU nano 7.2 lab6-3.asm
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

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/3, EDX=остаток от деления

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

```

Рис. 1.15: Обновленный код

Обновленный листинг кода:

```

#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

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'

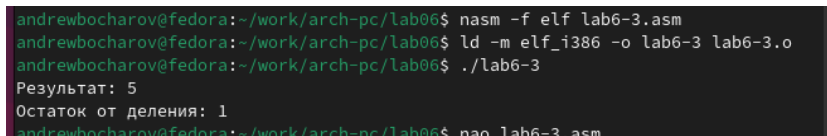
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов

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

```

Создал исполняемый файл и запустил его. Результат совпадает с ожидаемым.  
(рис. 1.16).



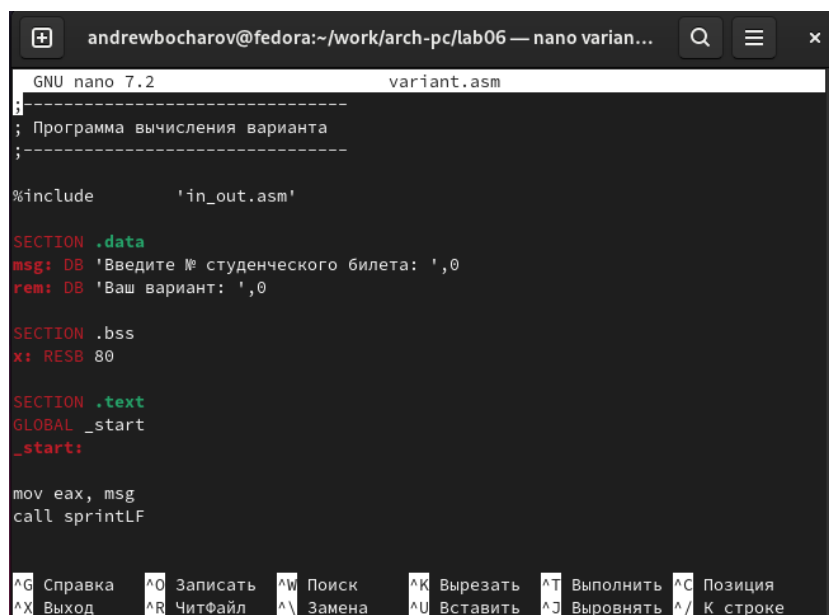
```

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

```

Рис. 1.16: Результат выполнения

Создал файл variant.asm и переписал в него листинг кода 6.4 (рис. 1.17).



```
andrewbocharov@fedora:~/work/arch-pc/lab06 — nano varian...
GNU nano 7.2 variant.asm
;-----
; 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

^G Справка  ^O Записать  ^W Поиск    ^K Вырезать  ^T Выполнить ^C Позиция
^X Выход    ^R ЧитФайл  ^\ Замена  ^U Вставить  ^J Выровнять ^/_ К строке
```

Рис. 1.17: Работает файла

Листинг 6.4:

```
;-----
; 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

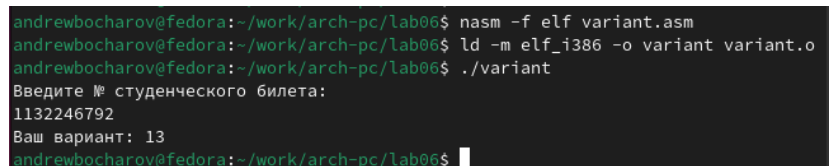
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx,edx
mov ebx,20
div ebx
inc edx

mov eax,rem
call sprintf
mov eax,edx
call iprintLF

call quit

```

Создал исполняемый файл и запустил его, ввел номер своего студенческого билета, в ответ выведен номер моего варианта. (рис. 1.18).



```

andrewbocharov@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
andrewbocharov@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
andrewbocharov@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246792
Ваш вариант: 13
andrewbocharov@fedora:~/work/arch-pc/lab06$

```

Рис. 1.18: Номер варианта

Ответы на вопросы: 1. За вывод на экран сообщения 'Ваш вариант:' отвечают строки `mov eax,rem` `call sprint` 2. Инструкции `mov ecx, x` `mov edx, 80` `call sread` отвечают за сохранение вввода в консоль в `x`

3 Инструкция "call atoi" переводит строку из консоли в число

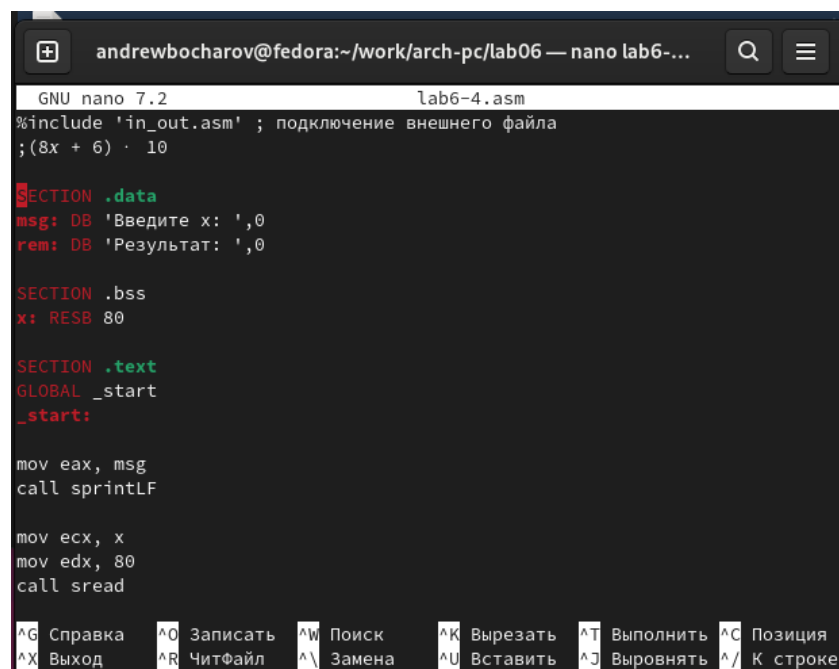
4 За вычисления варианта отвечают строки `mov ebx,20` `div ebx` `inc edx` 5 Остаток от деления при выполнении инструкции `div ebx` записывается в регистр `edx`

6 Инструкция `inc edx` увеличивает значения числа в регистре на 1

7 За вывод на экран результата вычислений отвечают строки `mov eax,edx` `call iprintLF`

## 2 Выполнение самостоятельной работы

Создал новый файл и написал в него код отвечающий за вычисление формулы соответствующей моему варианту. (рис. 2.1).



```
GNU nano 7.2 lab6-4.asm
#include 'in_out.asm' ; подключение внешнего файла
;(8x + 6) · 10

SECTION .data
msg: DB 'Введите x: ',0
rem: DB 'Результат: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
call sread

^G Справка  ^O Записать  ^W Поиск    ^K Вырезать ^T Выполнить ^C Позиция
^X Выход    ^R ЧитФайл  ^\ Замена  ^U Вставить ^J Выровнять ^_ К строке
```

Рис. 2.1: Листинг кода

Листинг кода:

```
%include 'in_out.asm' ; подключение внешнего файла
;(8x + 6) · 10
```

```

SECTION .data
msg: DB 'Введите x: ',0
rem: DB 'Результат: ',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`

mov ebx,8 ; EBX=8
mul ebx; EAX=EAX*EBX

add eax, 6 ; EAX=EAX+6
mov ebx,10 ; EBX=10
mul ebx ; EAX=EAX*EBX, EBX=Произведение
mov edx, eax

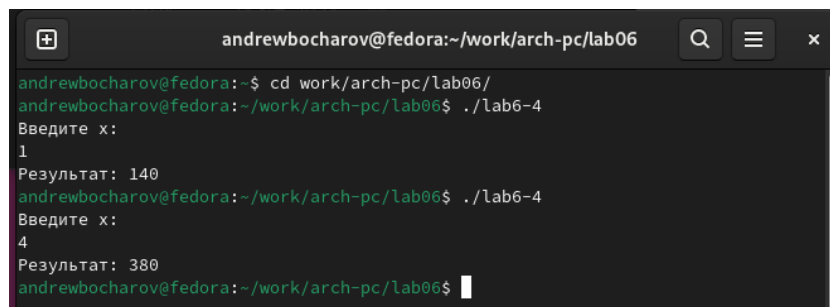
```



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

call quit ; вызов подпрограммы завершения
```

Сделал трансляцию, компоновку и запустил и проверил код. Код выводит правильные ответы, полученные аналитическим методом. (рис. 2.2).



```
andrewbocharov@fedora:~/work/arch-pc/lab06
andrewbocharov@fedora:~$ cd work/arch-pc/lab06/
andrewbocharov@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
1
Результат: 140
andrewbocharov@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
4
Результат: 380
andrewbocharov@fedora:~/work/arch-pc/lab06$
```

Рис. 2.2: Результат выполнения

Создал копию файла lab5-2.asm и внес в него изменения, что бы после ввода данных, они выводились на экран с использованием команд из in\_out.asm. Листинг кода:

## **3 Выводы**

Выполнив данную лабораторную работу, я освоил арифметические инструкции языка ассемблера NASM