

# **Отчёта по лабораторной работе 6**

**Освоение арифметических инструкций языка ассемблера NASM.**

Ягмыров Сохбет

# 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>Выводы</b>	<b>21</b>

## Список иллюстраций

2.1	Пример программы . . . . .	6
2.2	Работа программы . . . . .	7
2.3	Пример программы . . . . .	7
2.4	Работа программы . . . . .	8
2.5	Пример программы . . . . .	9
2.6	Работа программы . . . . .	9
2.7	Пример программы . . . . .	10
2.8	Работа программы . . . . .	11
2.9	Работа программы . . . . .	11
2.10	Пример программы . . . . .	12
2.11	Работа программы . . . . .	13
2.12	Пример программы . . . . .	14
2.13	Работа программы . . . . .	15
2.14	Пример программы . . . . .	16
2.15	Работа программы . . . . .	17
2.16	Пример программы . . . . .	19
2.17	Работа программы . . . . .	20

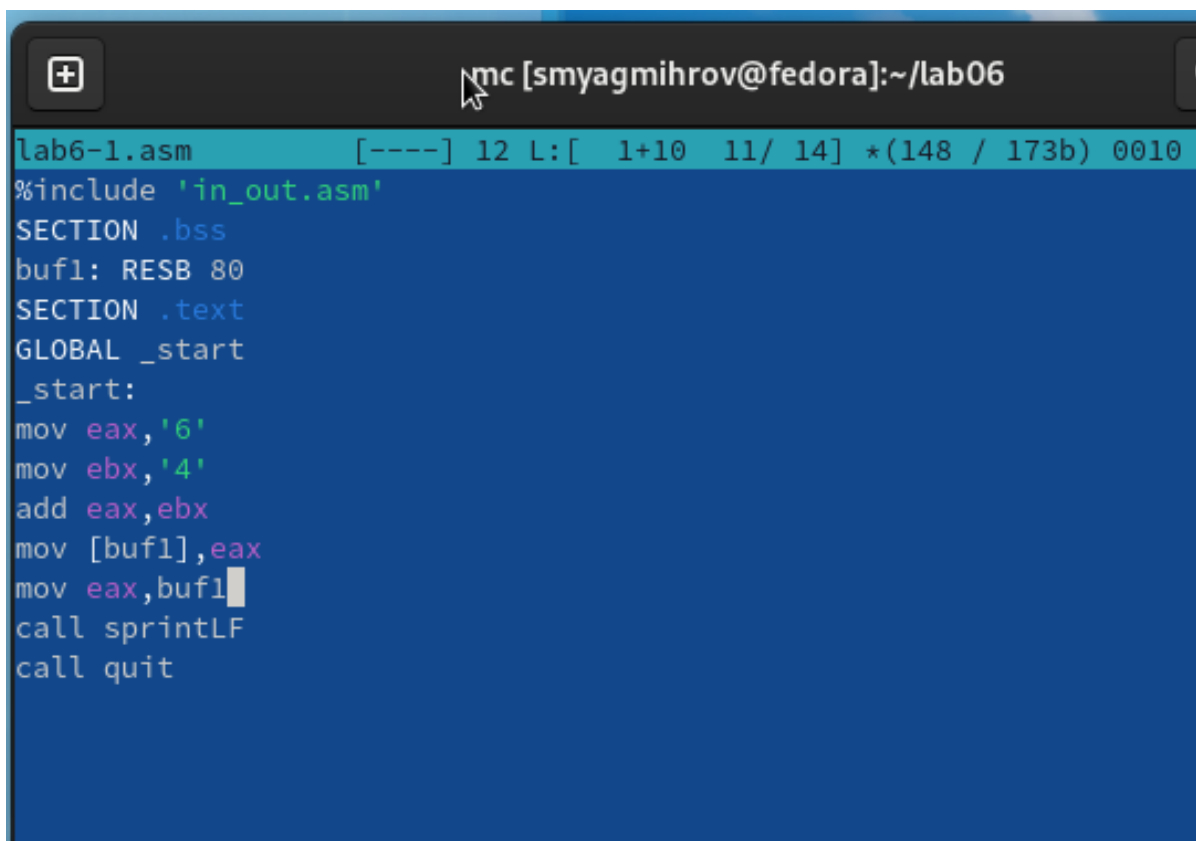
## Список таблиц

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

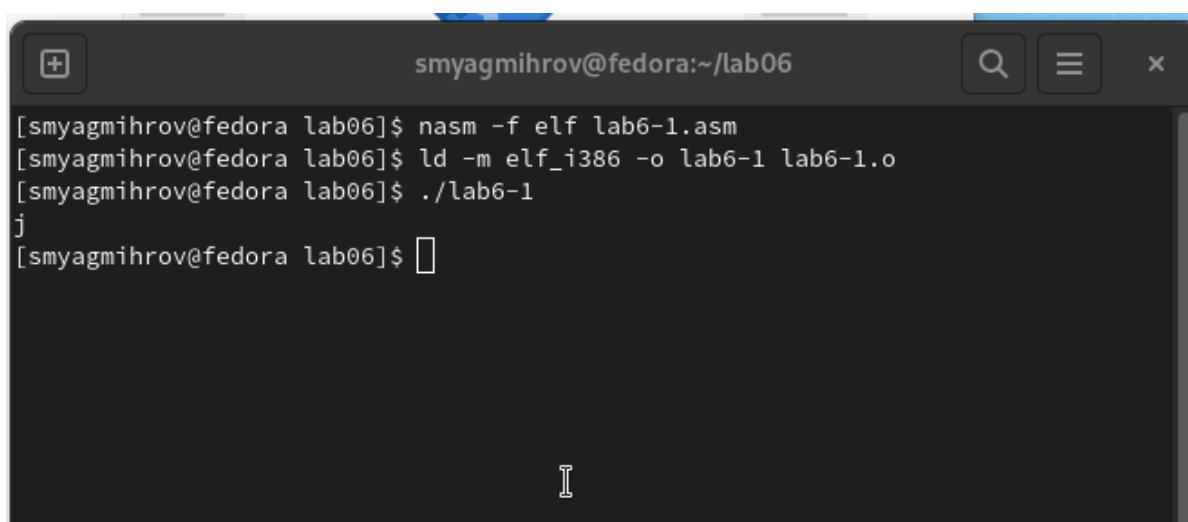
## 2 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax. (рис. 2.1, 2.2)



```
lab6-1.asm [----] 12 L: [ 1+10 11/ 14] *(148 / 173b) 0010
#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
```

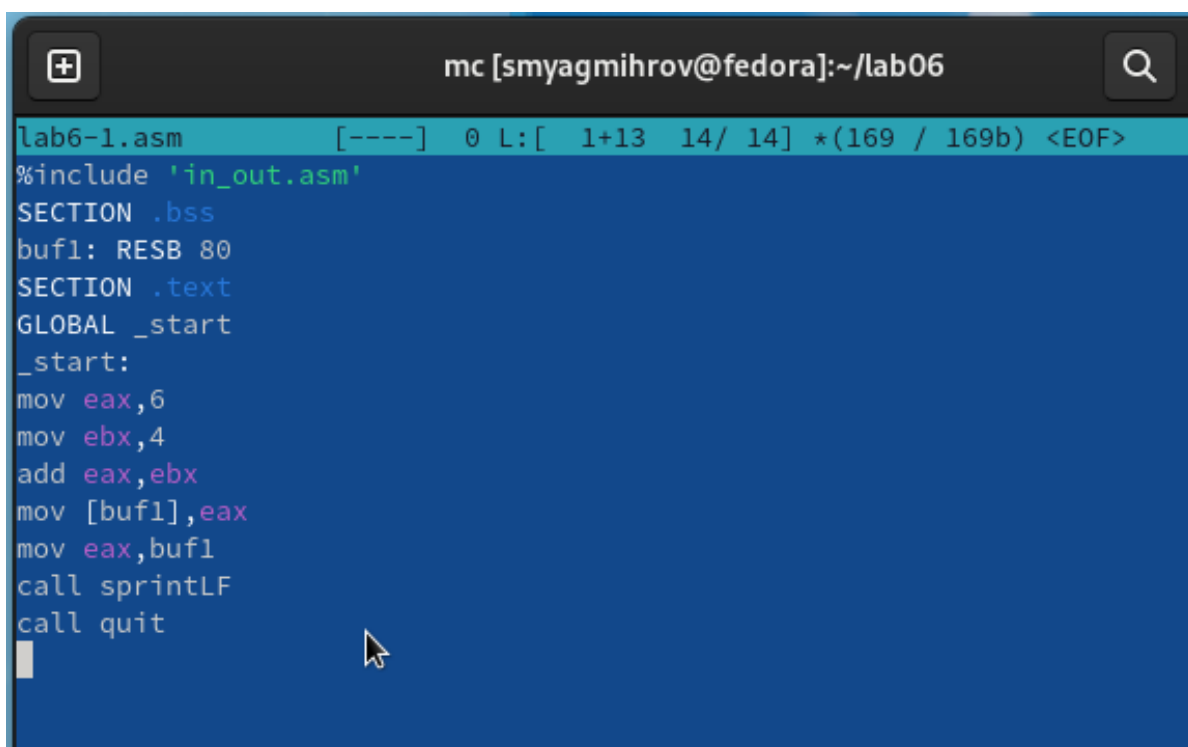
Рис. 2.1: Пример программы

A terminal window with a dark background. The title bar shows 'smyagmihrov@fedora:~/lab06'. The terminal contains the following commands and output:

```
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-1.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[smyagmihrov@fedora lab06]$ ./lab6-1
j
[smyagmihrov@fedora lab06]$
```

Рис. 2.2: Работа программы

3. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 1) следующим образом: (рис. 2.3, 2.4)

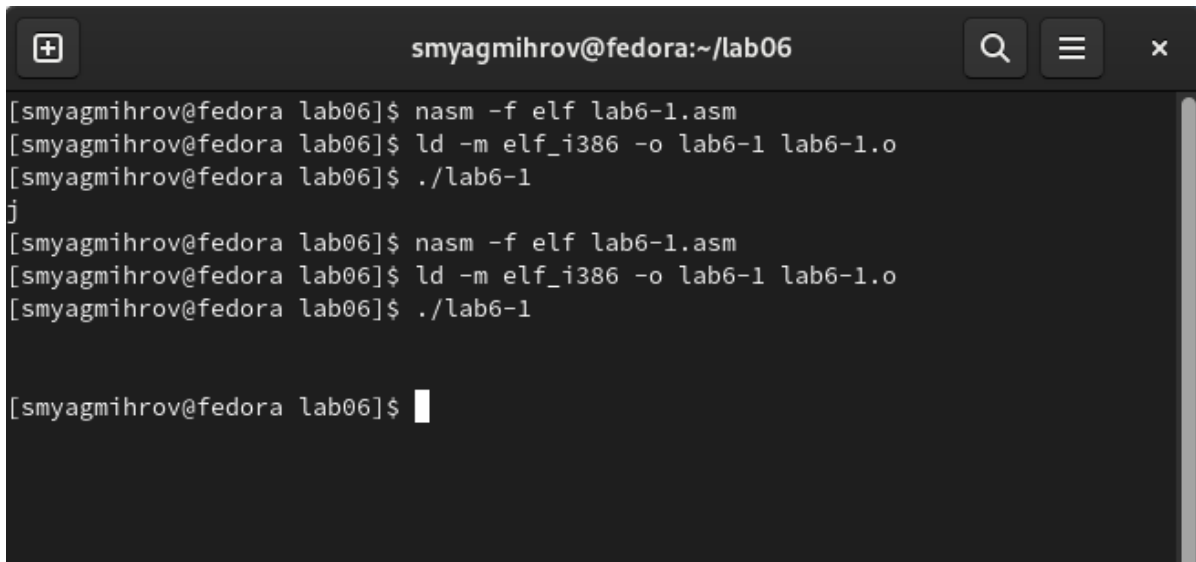
A terminal window with a dark blue background. The title bar shows 'mc [smyagmihrov@fedora]:~/lab06'. The terminal displays the assembly code for 'lab6-1.asm'. The first line is highlighted in light blue:

```
lab6-1.asm [----] 0 L: [ 1+13 14/ 14] *(169 / 169b) <EOF>
```

The rest of the code is as follows:

```
%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. 2.3: Пример программы

A terminal window titled 'smyagmihrov@fedora:~/lab06' with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

```
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-1.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[smyagmihrov@fedora lab06]$ ./lab6-1
j
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-1.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[smyagmihrov@fedora lab06]$ ./lab6-1

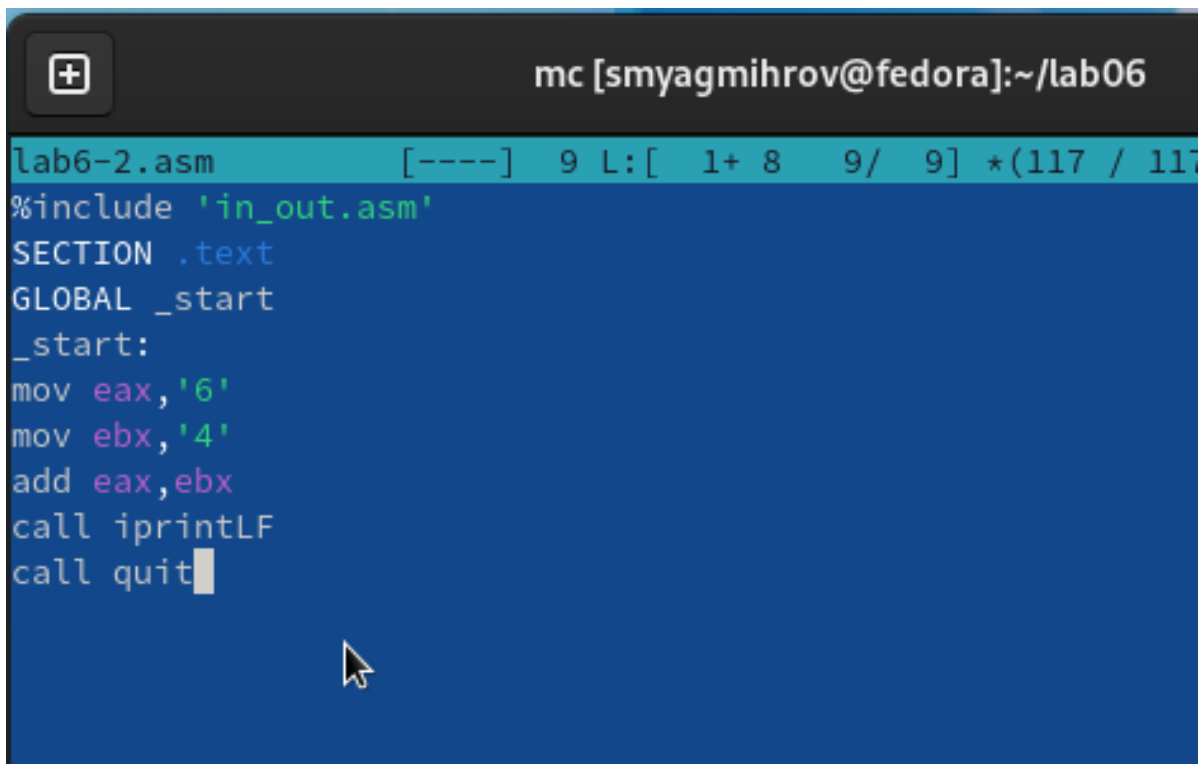
[smyagmihrov@fedora lab06]$
```

Рис. 2.4: Работа программы

Никакой символ не виден, но он есть. Это возврат каретки LF.

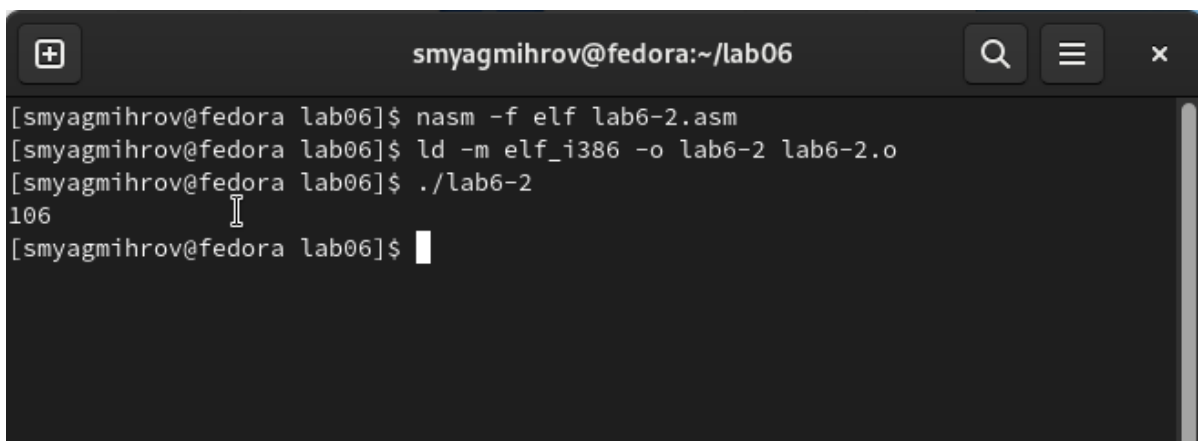
4. Как отмечалось выше, для работы с числами в файле `in_out.asm` реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 7.1 с использованием этих функций. (рис. 2.5, 2.6)





```
mc [smyagmihrov@fedora]:~/lab06
lab6-2.asm [-----] 9 L: [ 1+ 8 9/ 9] *(117 / 117)
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 2.5: Пример программы



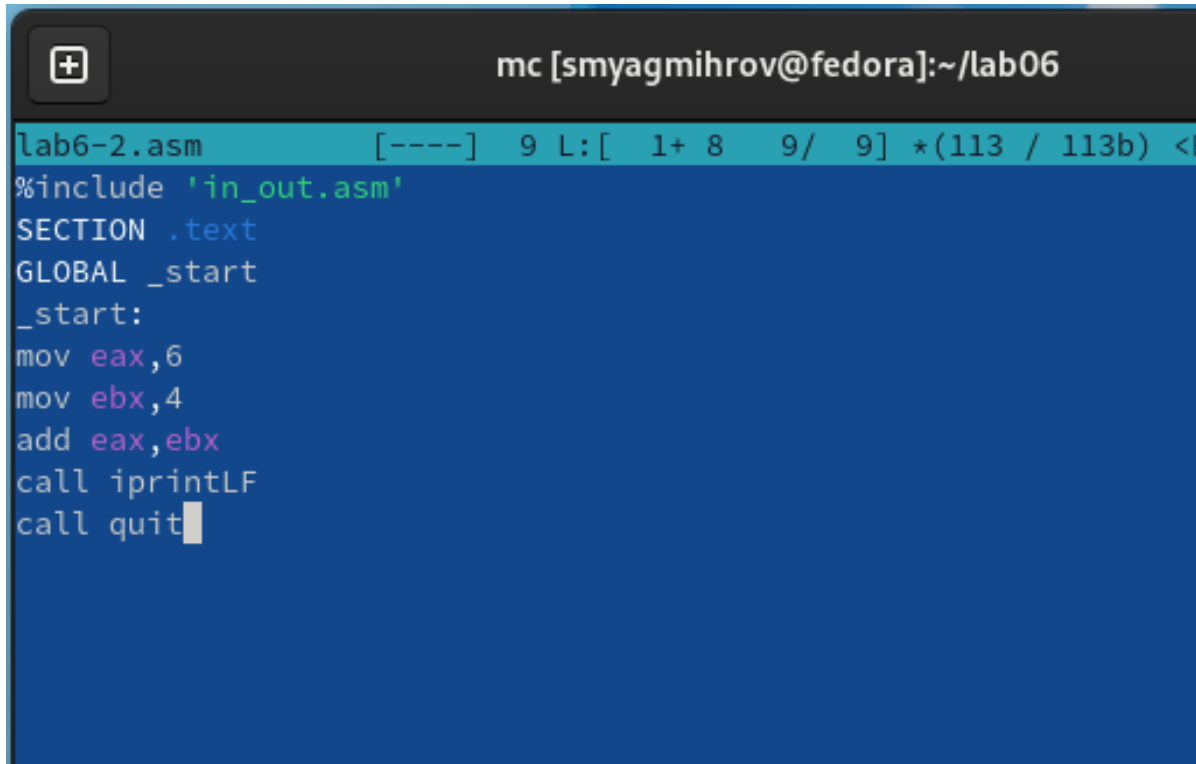
```
smyagmihrov@fedora:~/lab06
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-2.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[smyagmihrov@fedora lab06]$ ./lab6-2
106
[smyagmihrov@fedora lab06]$
```

Рис. 2.6: Работа программы

В результате работы программы мы получим число 106. В данном случае, как и в первом, команда `add` складывает коды символов '6' и '4' ( $54+52=106$ ). Однако, в отличие от программы из листинга 7.1, функция `iprintLF` позволяет вывести число, а не символ, кодом которого является это число.

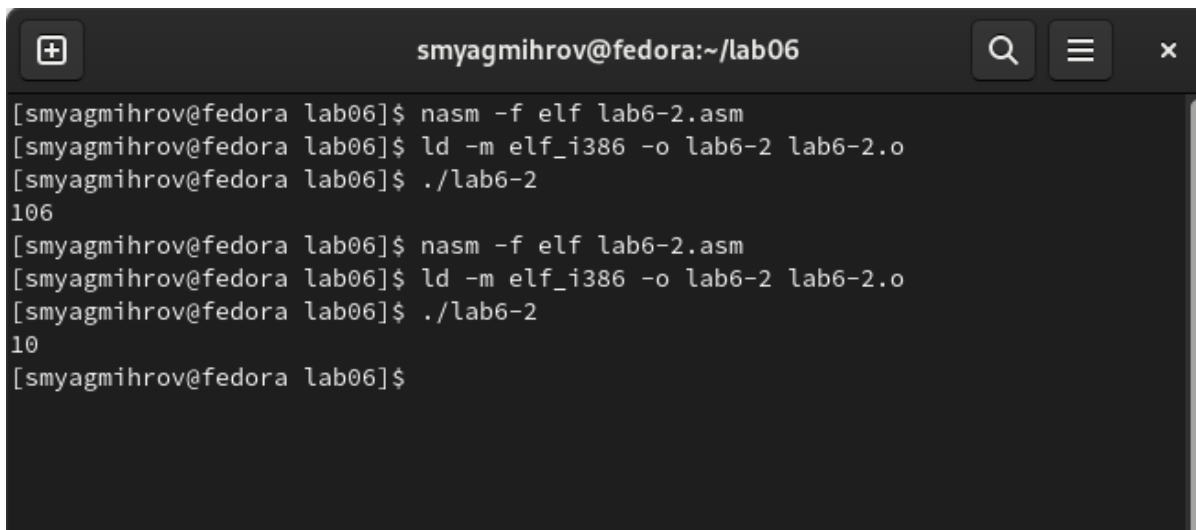
5. Аналогично предыдущему примеру изменим символы на числа. (рис. 2.7, 2.8)

Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы? – получили число 10



```
mc [smyagmihrov@fedora]:~/lab06
lab6-2.asm [----] 9 L:[ 1+ 8 9/ 9] *(113 / 113b) <f
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

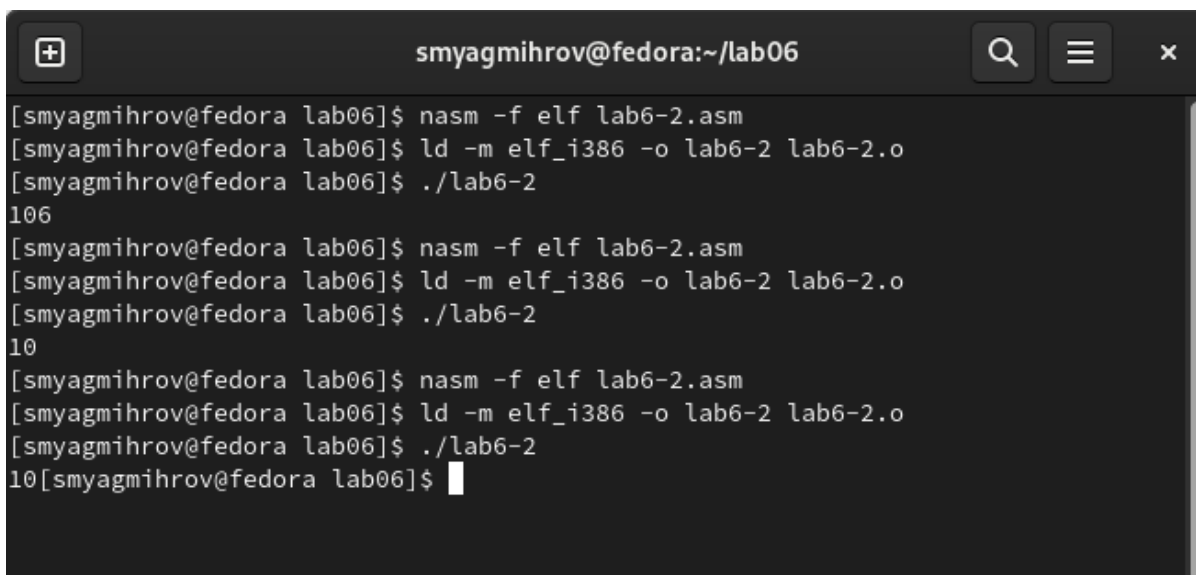
Рис. 2.7: Пример программы

A terminal window titled 'smyagmihrov@fedora:~/lab06' with search, menu, and close icons. It shows the following commands and output:

```
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-2.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[smyagmihrov@fedora lab06]$ ./lab6-2
106
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-2.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[smyagmihrov@fedora lab06]$ ./lab6-2
10
[smyagmihrov@fedora lab06]$
```

Рис. 2.8: Работа программы

Замените функцию `iprintLF` на `iprint`. Создайте исполняемый файл и запустите его. Чем отличается вывод функций `iprintLF` и `iprint`? - Вывод отличается что нет переноса строки. (рис. 2.9)

A terminal window titled 'smyagmihrov@fedora:~/lab06' with search, menu, and close icons. It shows the following commands and output:

```
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-2.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[smyagmihrov@fedora lab06]$ ./lab6-2
106
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-2.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[smyagmihrov@fedora lab06]$ ./lab6-2
10
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-2.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[smyagmihrov@fedora lab06]$ ./lab6-2
10[smyagmihrov@fedora lab06]$ █
```

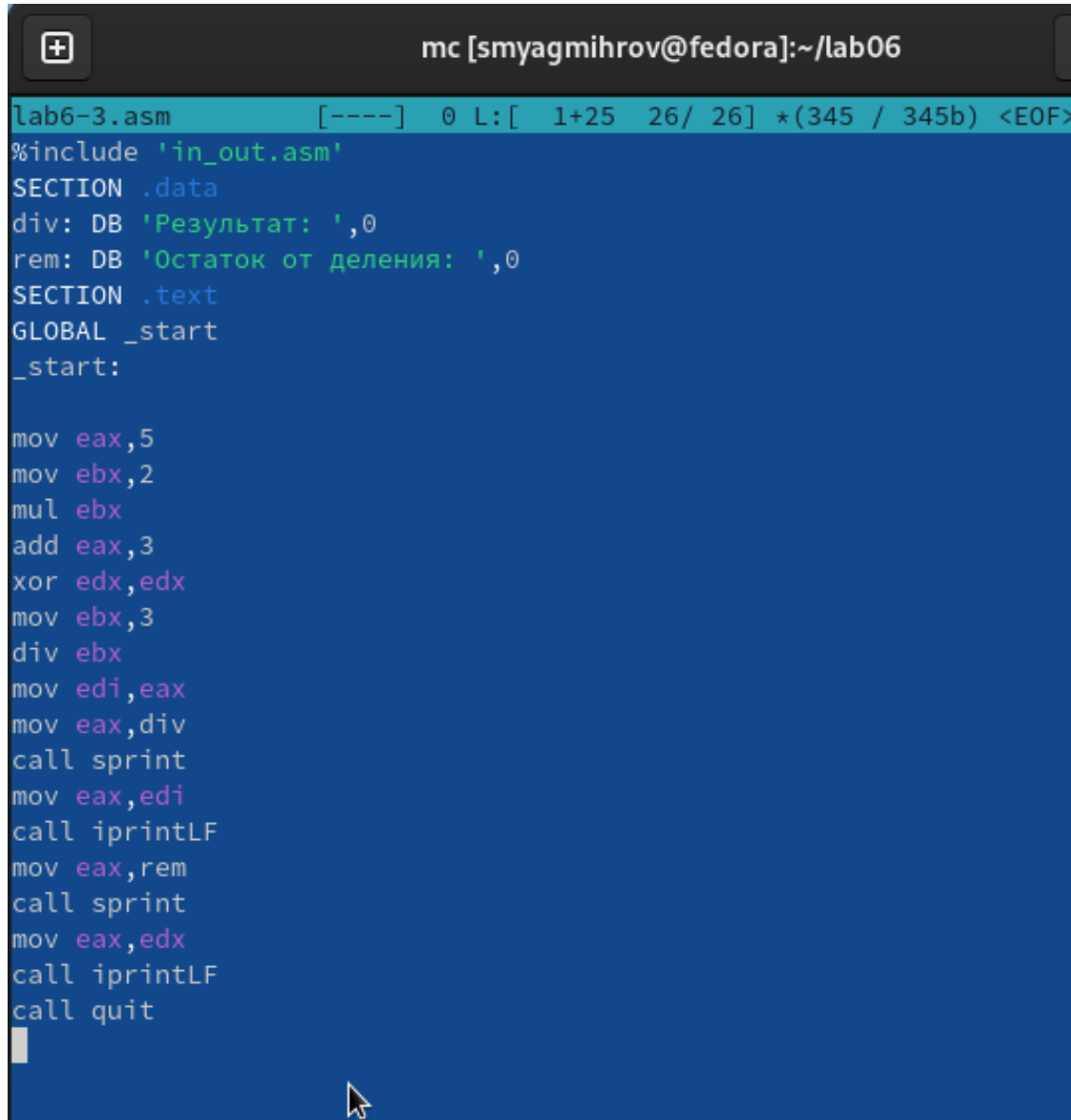
Рис. 2.9: Работа программы

6. В качестве примера выполнения арифметических операций в NASM приве-

дем программу вычисления арифметического выражения

$$f(x) = (5 * 2 + 3) / 3$$

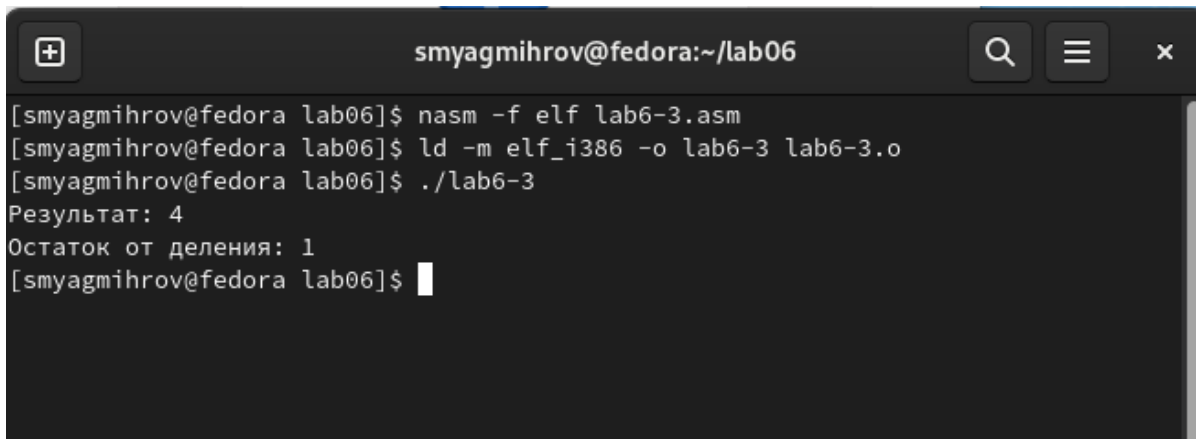
. (рис. 2.10, рис. 2.11)



```
mc [smyagmihrov@fedora]:~/lab06
lab6-3.asm [----] 0 L: [ 1+25 26/ 26] *(345 / 345b) <EOF>
#include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:

mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx
mov edi,eax
mov eax,div
call sprint
mov eax,edi
call iprintLF
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 2.10: Пример программы



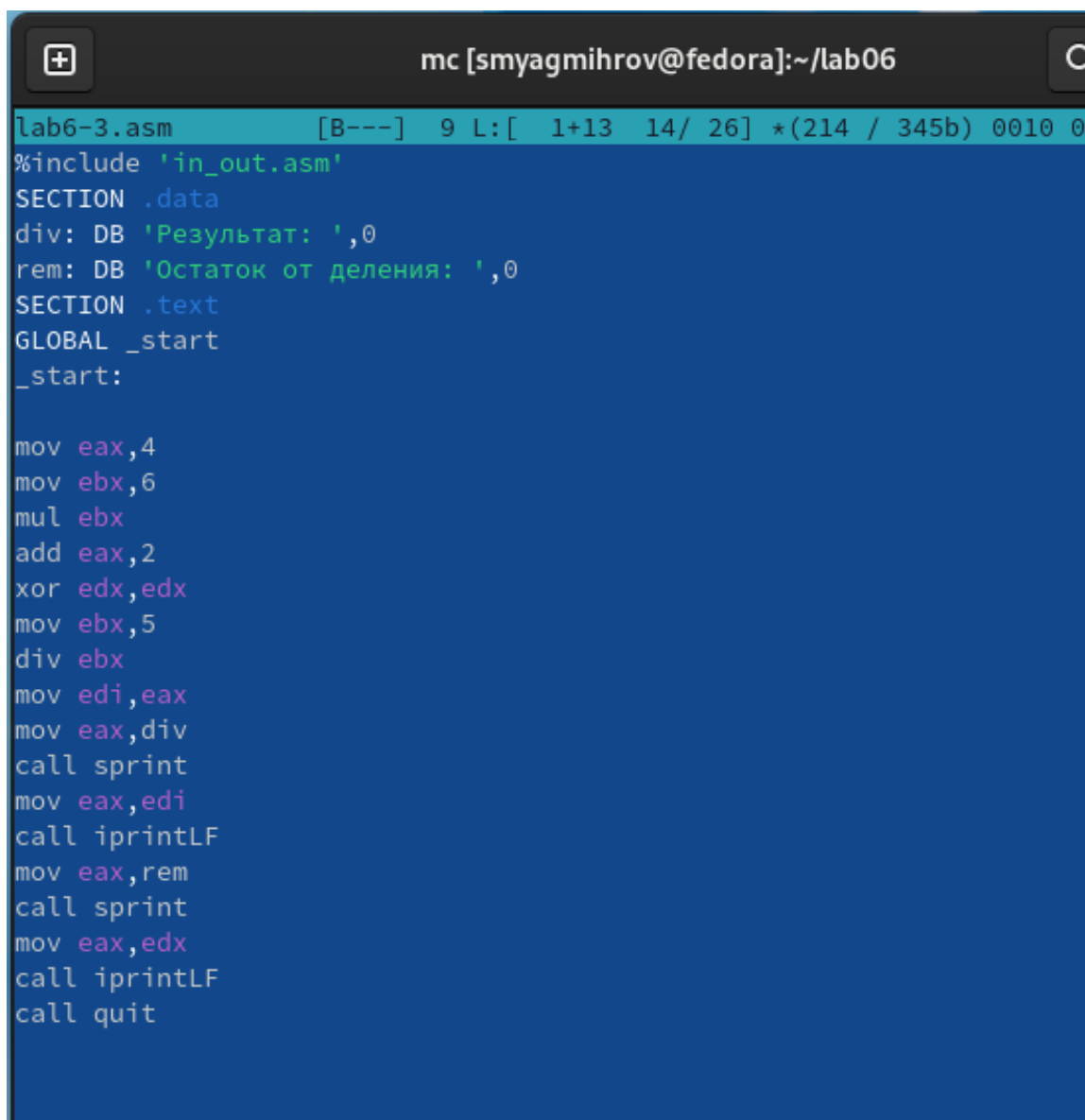
```
smyagmihrov@fedora:~/lab06
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-3.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[smyagmihrov@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
[smyagmihrov@fedora lab06]$
```

Рис. 2.11: Работа программы

Измените текст программы для вычисления выражения

$$f(x) = (4 * 6 + 2) / 5$$

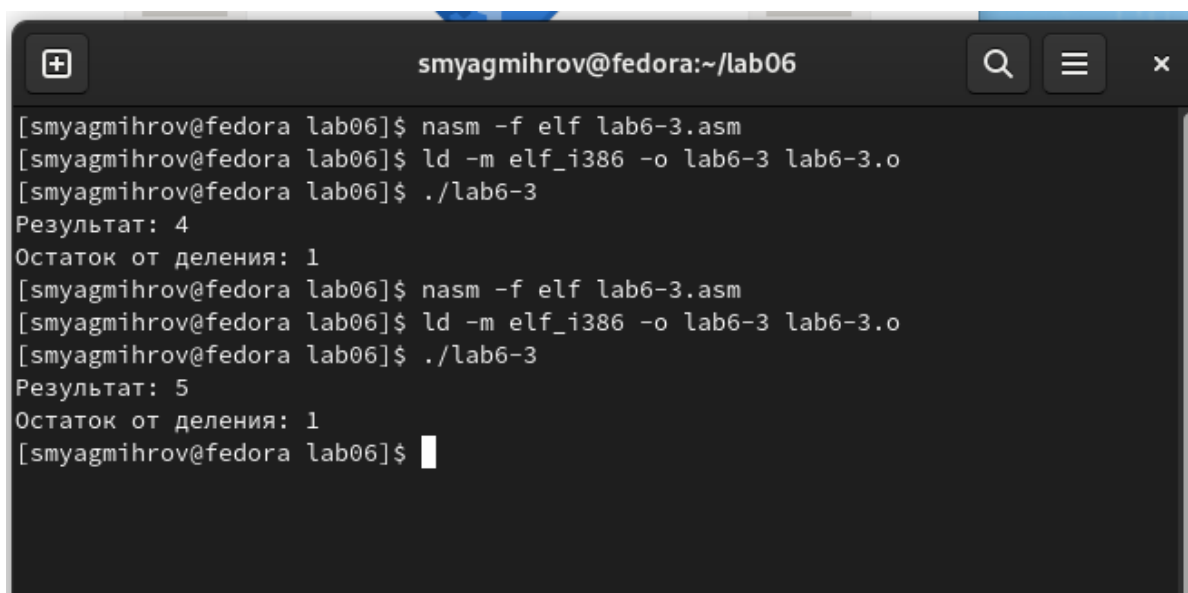
. Создайте исполняемый файл и проверьте его работу. (рис. 2.12, рис. 2.13)



```
mc [smyagmihrov@fedora]:~/lab06
lab6-3.asm [B---] 9 L:[ 1+13 14/ 26] *(214 / 345b) 0010 0
#include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx
mov edi,eax
mov eax,div
call sprint
mov eax,edi
call iprintLF
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 2.12: Пример программы

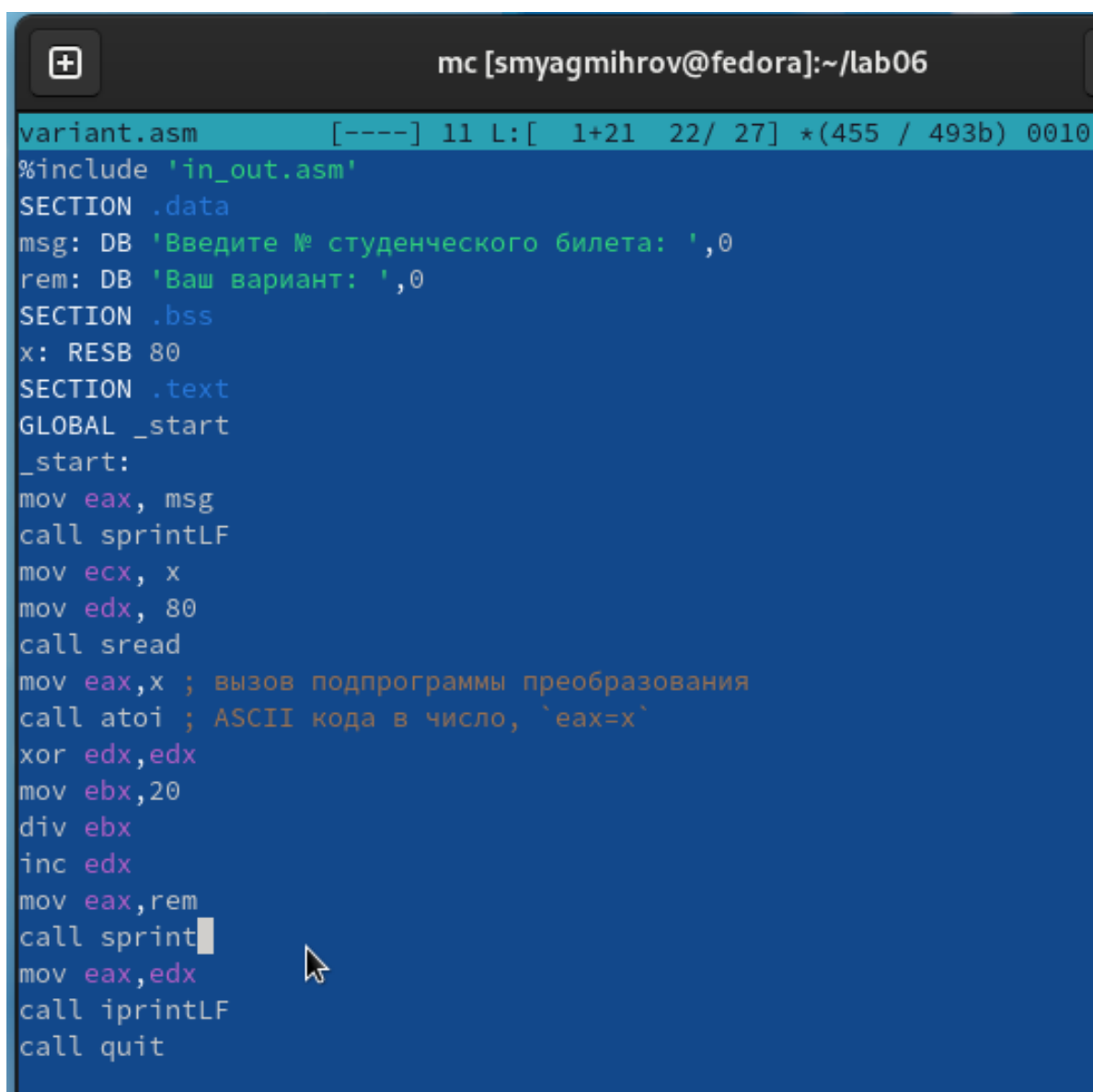


The image shows a terminal window with a dark background. The title bar at the top reads 'smyagmihrov@fedora:~/lab06'. The terminal contains the following text:

```
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-3.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[smyagmihrov@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
[smyagmihrov@fedora lab06]$ nasm -f elf lab6-3.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[smyagmihrov@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[smyagmihrov@fedora lab06]$
```

Рис. 2.13: Работа программы

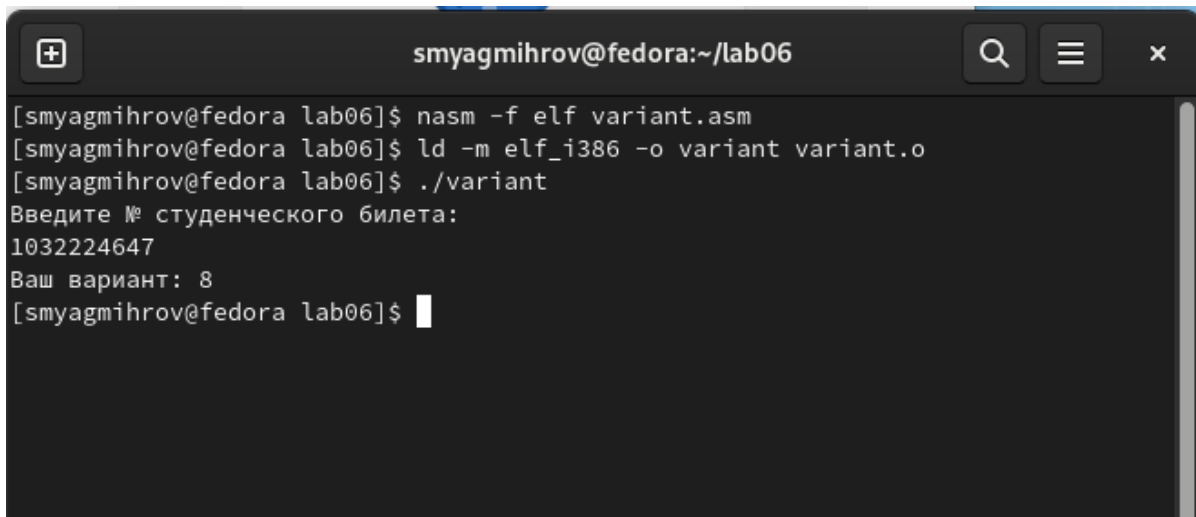
7. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму: (рис. 2.14, рис. 2.15)



```
variant.asm      [----] 11 L:[ 1+21 22/ 27] *(455 / 493b) 0010
#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
```

Рис. 2.14: Пример программы





```
[smyagmihrov@fedora lab06]$ nasm -f elf variant.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[smyagmihrov@fedora lab06]$ ./variant
Введите № студенческого билета:
1032224647
Ваш вариант: 8
[smyagmihrov@fedora lab06]$
```

Рис. 2.15: Работа программы

- Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? – `mov eax,rem` – перекладывает в регистр значение переменной с фразой ‘Ваш вариант:’ `call sprint` – вызов подпрограммы вывода строки
- Для чего используются следующие инструкции? `nasm mov ecx, x mov edx, 80 call sread`

Считывает значение студбилета в переменную X из консоли

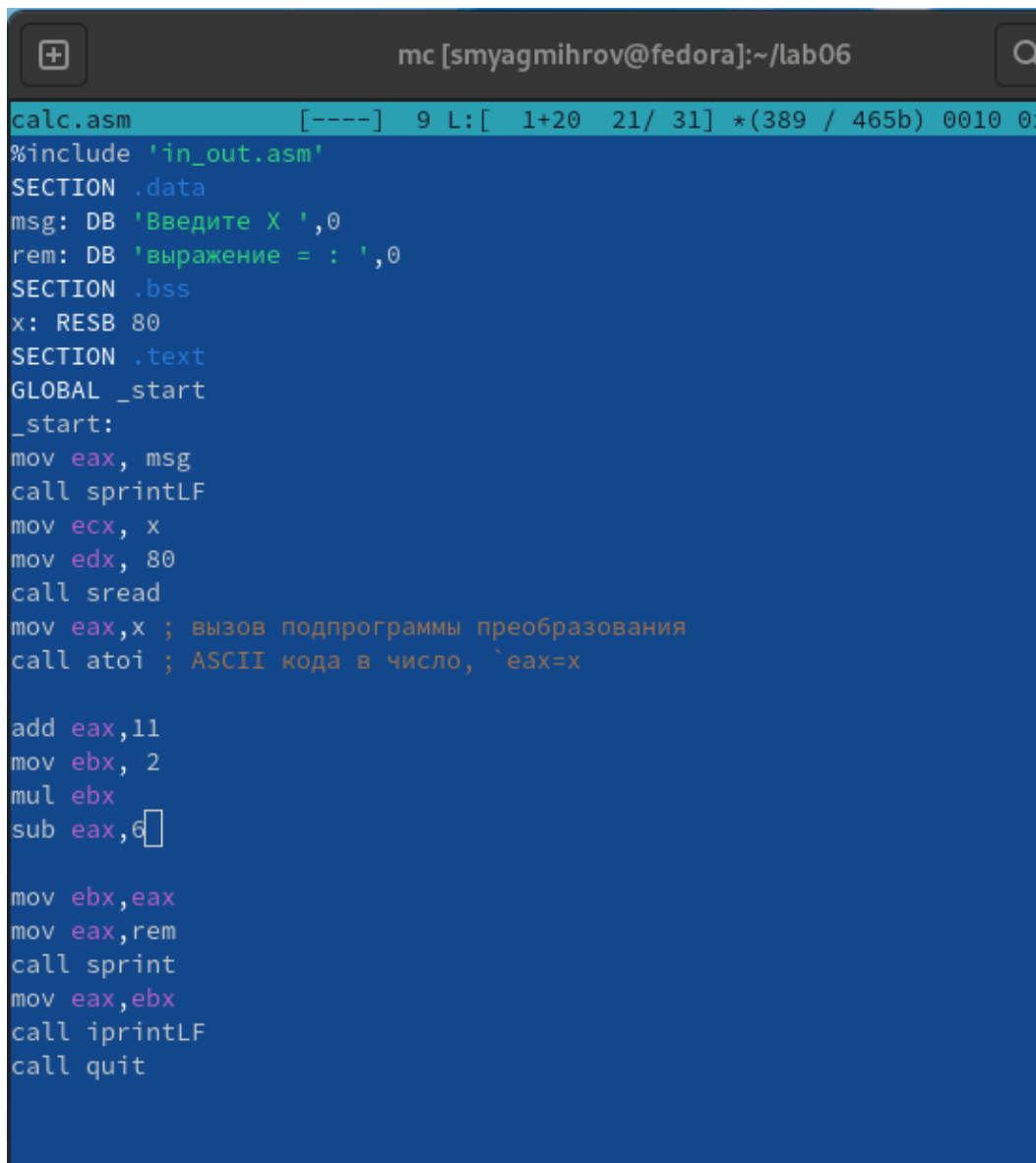
- Для чего используется инструкция “`call atoi`”? – эта подпрограмма переводит введенные символы в числовой формат
- Какие строки листинга 7.4 отвечают за вычисления варианта? `xor edx,edx mov ebx,20 div ebx`
- В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”? 1 байт AH 2 байта DX 4 байта EDX – наш случай
- Для чего используется инструкция “`inc edx`”? по формуле вычисления варианта нужно прибавить единицу

- Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений `mov eax,edx` – результат перекладывается в регистр `eax` `call iprintLF` – вызов подпрограммы вывода
8. Написать программу вычисления выражения  $y = f(x)$ . Программа должна выводить выражение для вычисления, выводить запрос на ввод значения  $x$ , вычислять заданное выражение в зависимости от введенного  $x$ , выводить результат вычислений. Вид функции  $f(x)$  выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений  $x_1$  и  $x_2$  из 6.3. (рис. 2.16, рис. 2.17)

Получили вариант 8 -

$$(11 + x) * 2 - 6$$

для  $x=1$  и  $9$

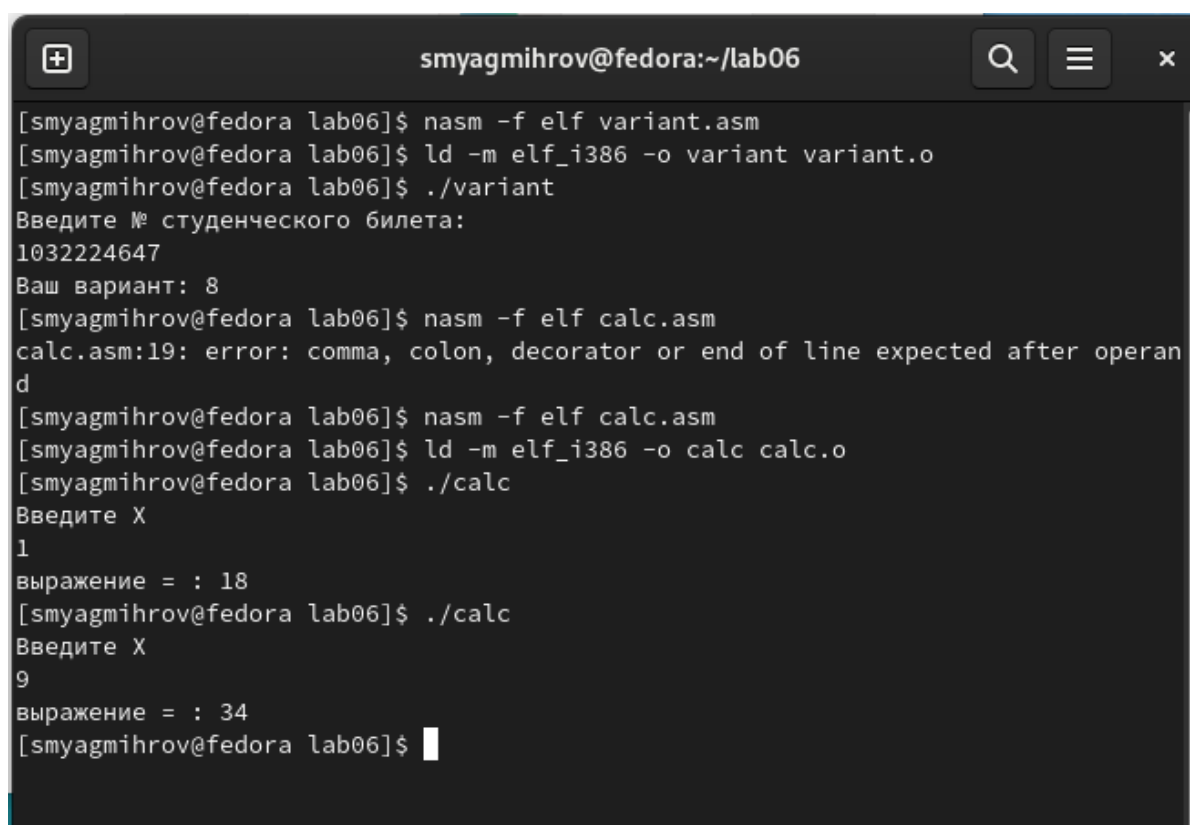


```
mc [smyagmihrov@fedora]:~/lab06
calc.asm [----] 9 L:[ 1+20 21/ 31] *(389 / 465b) 0010 0
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите X ',0
rem: DB 'выражение = : ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x

add eax, 11
mov ebx, 2
mul ebx
sub eax, 6

mov ebx, eax
mov eax, rem
call sprintf
mov eax, ebx
call iprintLF
call quit
```

Рис. 2.16: Пример программы

A terminal window titled 'smyagmihrov@fedora:~/lab06' with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

```
[smyagmihrov@fedora lab06]$ nasm -f elf variant.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[smyagmihrov@fedora lab06]$ ./variant
Введите № студенческого билета:
1032224647
Ваш вариант: 8
[smyagmihrov@fedora lab06]$ nasm -f elf calc.asm
calc.asm:19: error: comma, colon, decorator or end of line expected after operand
[smyagmihrov@fedora lab06]$ nasm -f elf calc.asm
[smyagmihrov@fedora lab06]$ ld -m elf_i386 -o calc calc.o
[smyagmihrov@fedora lab06]$ ./calc
Введите X
1
выражение = : 18
[smyagmihrov@fedora lab06]$ ./calc
Введите X
9
выражение = : 34
[smyagmihrov@fedora lab06]$
```

Рис. 2.17: Работа программы

## **3 Выводы**

Изучили работу с арифметическими операциями