

# **Лабораторная работа №6**

**Арифметические операции в NASM**

Борисенкова София Павловна

# 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>Выполнение задания для самостоятельной работы</b>	<b>17</b>
<b>4</b>	<b>Выводы</b>	<b>19</b>

# Список иллюстраций

2.1	Создание рабочей директории и файла lab6-1.asm . . . . .	6
2.2	Запись кода из листинга в файл lab6-1.asm . . . . .	6
2.3	Копирование файла in_out.asm в рабочую директорию . . . . .	7
2.4	Запуск исполняемого файла и результат вывода . . . . .	7
2.5	Редактирование файла . . . . .	8
2.6	Запуск исполняемого файла и результат вывода . . . . .	8
2.7	Создание файла lab6-2.asm . . . . .	9
2.8	Запись кода в файл lab6-2.asm . . . . .	9
2.9	Запуск исполняемого файла и результат вывода . . . . .	10
2.10	Изменение файла lab6-2.asm . . . . .	10
2.11	Сборка исполняемого файла и результат работы программы . . . .	11
2.12	Редактирование файла lab6-2.asm . . . . .	11
2.13	Сборка и результат работы отредактированного файла . . . . .	12
2.14	Создание файла lab6-3.asm . . . . .	12
2.15	Вставка кода из листинга в файл . . . . .	13
2.16	Сборка файла lab6-3.asm и результат его работы . . . . .	13
2.17	Редактирование файла lab6-3.asm . . . . .	14
2.18	Сборка файла lab6-3.asm и результат его работы . . . . .	14
2.19	Создание файла variant.asm для вычисления варианта для самостоятельной работы . . . . .	15
2.20	Вставка кода в файл variant.asm . . . . .	15
2.21	Сборка и запуск программы variant . . . . .	16
3.1	Код требуемой программы . . . . .	17
3.2	Запуск программы и проверка её корректной работы . . . . .	18

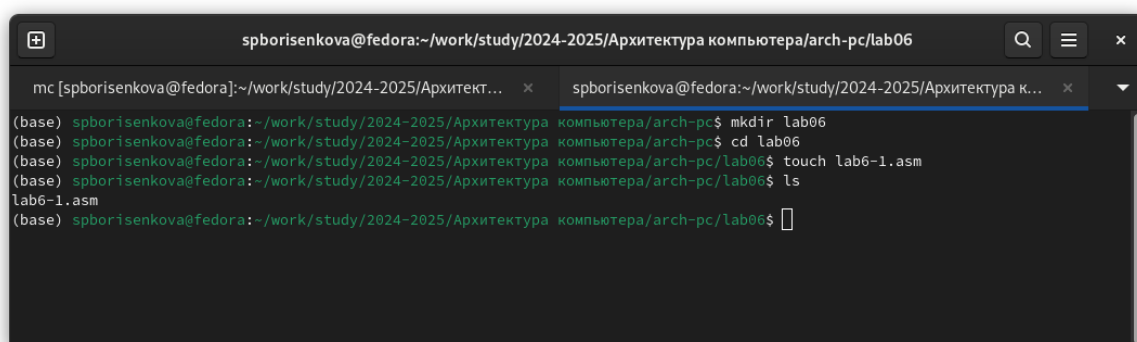
## **Список таблиц**

# 1 Цель работы

Познакомиться с базовыми инструкциями языка Ассемблер, отвечающими за основные арифметические операции

## 2 Выполнение лабораторной работы

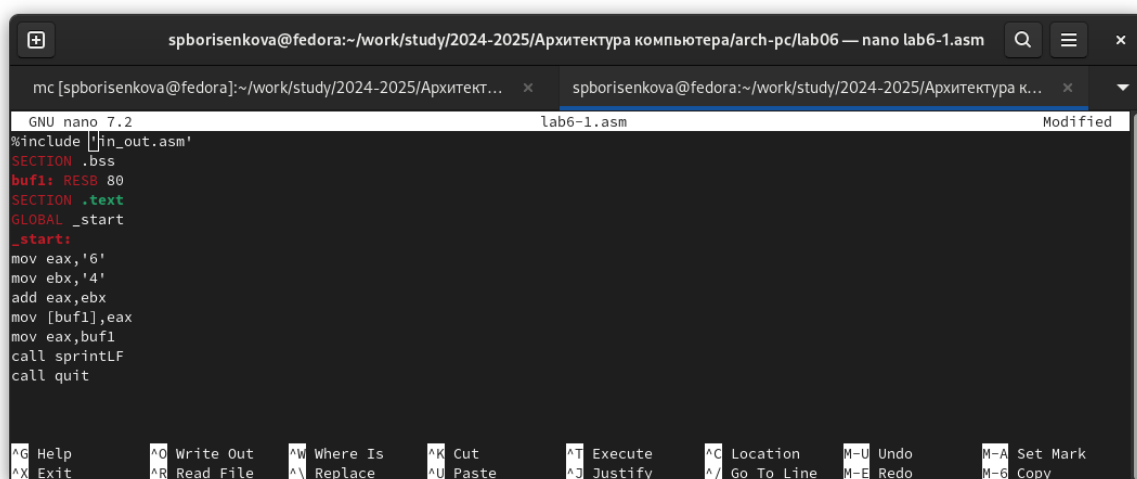
Для начала выполнения лабораторной работы необходимо создать папку рабочего каталога и файл lab6-1.asm (Рис. 2.1):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архитект... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура к... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ mkdir lab06
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd lab06
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-1.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
lab6-1.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.1: Создание рабочей директории и файла lab6-1.asm

Вставим в наш созданный файл код из листинга 6.1 (Рис. 2.2):

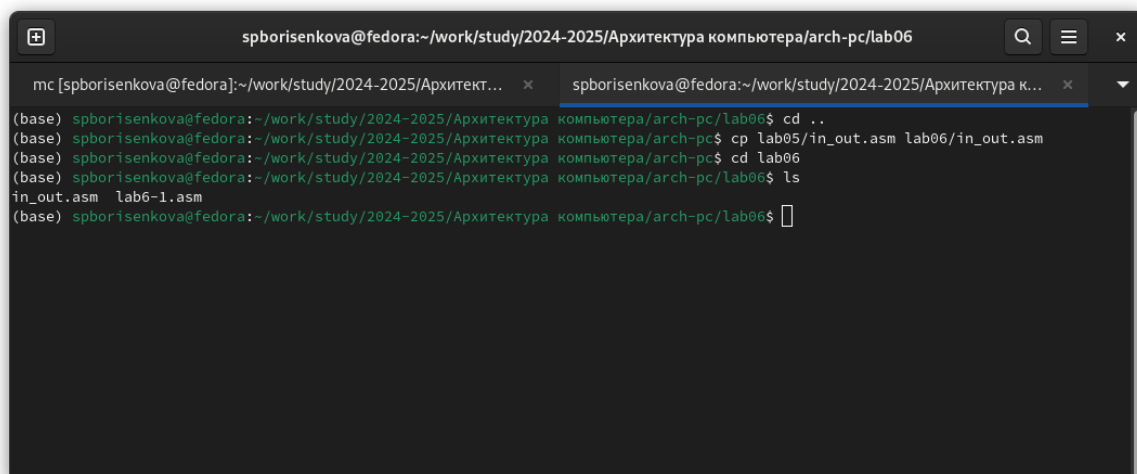


```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06 — nano lab6-1.asm
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архитект... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура к... x
GNU nano 7.2 lab6-1.asm Modified
%include "in_out.asm"
SECTION .bss
buf1: 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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo      M-6 Copy
```

Рис. 2.2: Запись кода из листинга в файл lab6-1.asm

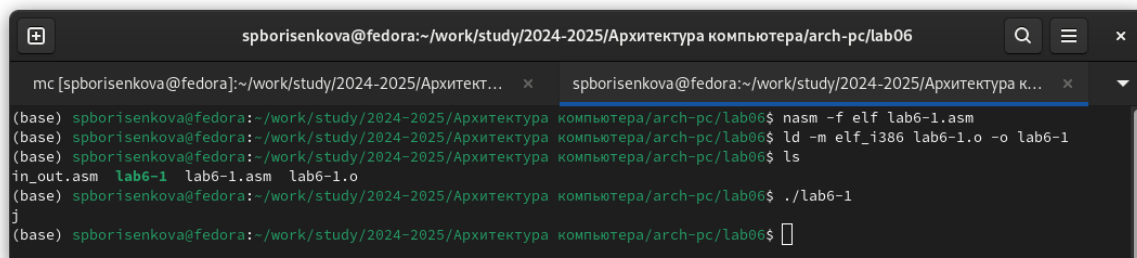
Перед сборкой файла стоит учесть, что он использует сторонний файл `in_out.asm`. Скопируем его из каталога пятой лабораторной работы (Рис. 2.3):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архитект... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура к... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ cd ..
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cp lab05/in_out.asm lab06/in_out.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd lab06
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.3: Копирование файла `in_out.asm` в рабочую директорию

Теперь соберём наш файл в исполняемое приложение. Запустим его и посмотрим на результат (Рис. 2.4):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архитект... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура к... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 lab6-1.o -o lab6-1
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-1
j
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.4: Запуск исполняемого файла и результат вывода

Программа выводит символ `j`, однако это неправильный вывод. Наша цель - сложить 6 и 4, и получить в выводе число 10. Изменим файл (Рис. 2.5):

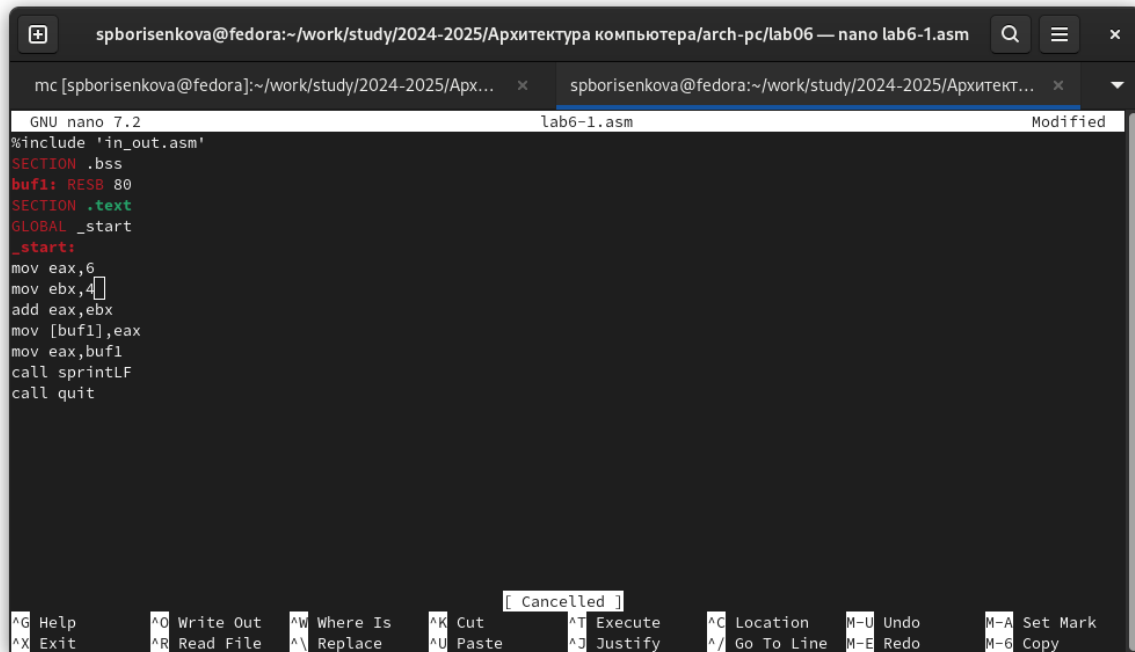


Рис. 2.5: Редактирование файла

Мы убрали кавычки у цифр, и теперь складываем уже не символы 6 и 4, а числа. Теперь попробуем собрать исполняемый файл также и запустим его (Рис. 2.6):

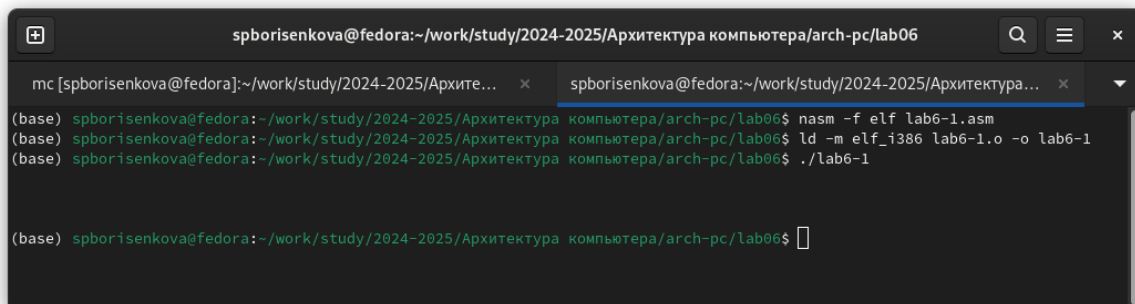
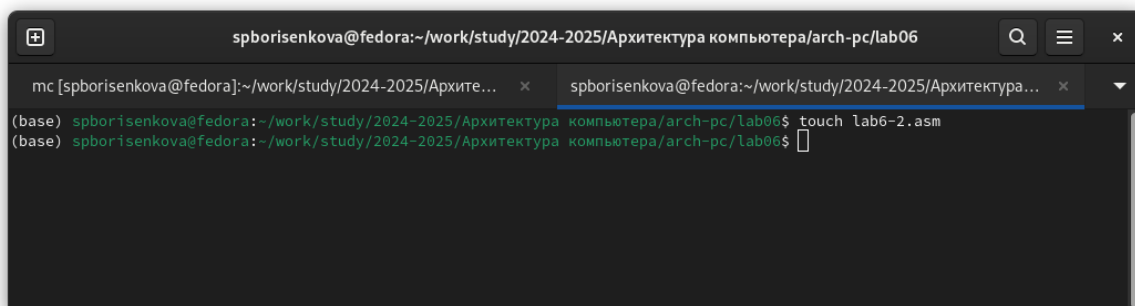


Рис. 2.6: Запуск исполняемого файла и результат вывода

Мы видим, что ничего не вывелось. Но так ли это? Когда мы вызываем команду `sprintf`, она выводит не число 10, а символ с номером 10. Посмотрим на таблицу ASCII и увидим, что символ под номером 10 это символ перевода строки. Имен-



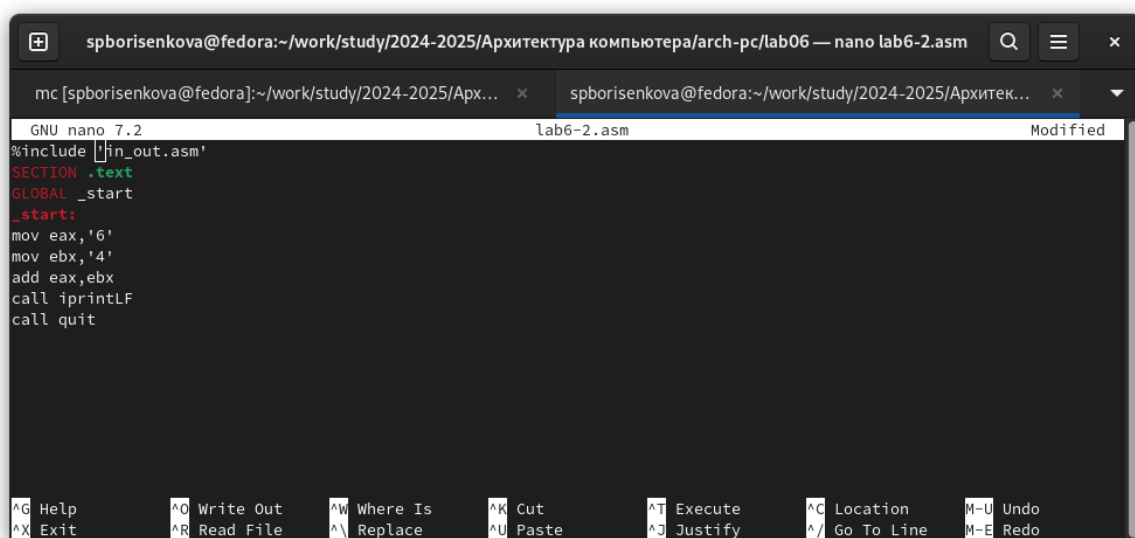
но поэтому мы его не видим, мы видим просто новую строку. Теперь создадим второй файл под названием lab6-2.asm (Рис. 2.7):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архите... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-2.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.7: Создание файла lab6-2.asm

Вставим в него код из листинга 6.2 (Рис. 2.8):

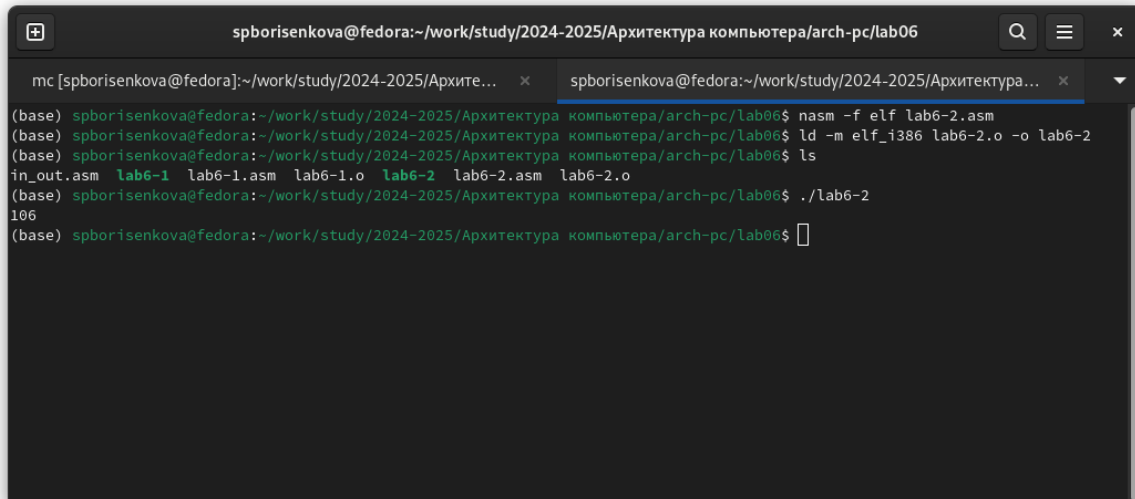


```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06 — nano lab6-2.asm
mc [spborisenkova@fedora]:~/work/study/2024-2025/Apx... x spborisenkova@fedora:~/work/study/2024-2025/Архитек... x
GNU nano 7.2 lab6-2.asm Modified
%include "in_out.asm"
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_/ Go To Line M-E Redo
```

Рис. 2.8: Запись кода в файл lab6-2.asm

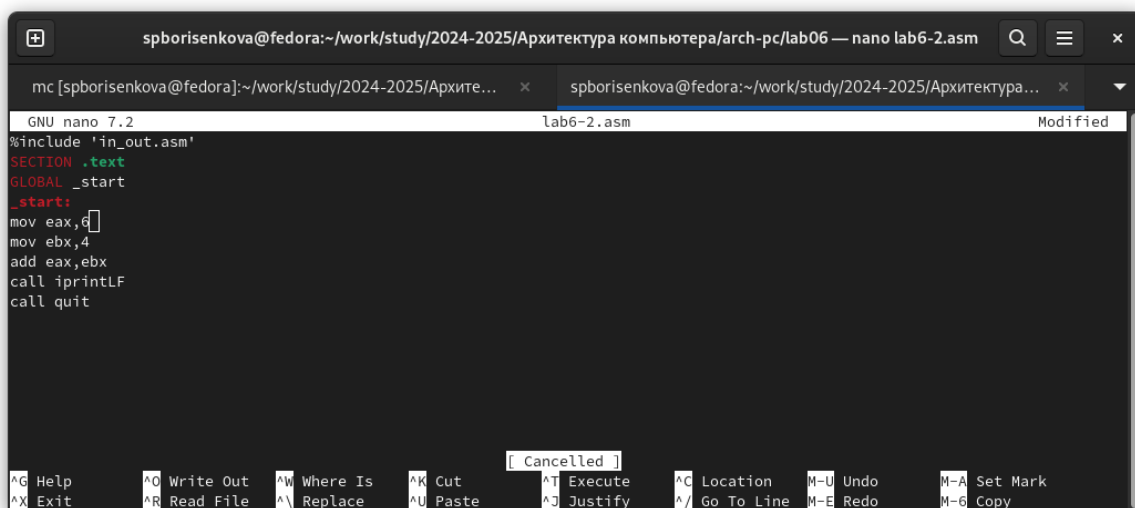
Основное отличие заключается в том, что вместо `sprintLF` используется `iprintLF`. Соберём файл и запустим его, чтобы посмотреть, как изменится вывод (Рис. 2.9):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora:~/work/study/2024-2025/Архите... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 lab6-2.o -o lab6-2
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2 lab6-2.asm lab6-2.o
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
106
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.9: Запуск исполняемого файла и результат вывода

Мы видим число 106. Так как цифры в коде указаны в кавычках, мы складываем их коды (54 и 52 в сумме дают 106). Теперь программа способна вывести число, а не символ ASCII с соответствующим номером. Теперь, если мы уберём кавычки у цифр, программа должна вывести 10. Убедимся в этом, сделав соответствующие изменения в коде (Рис. 2.10):

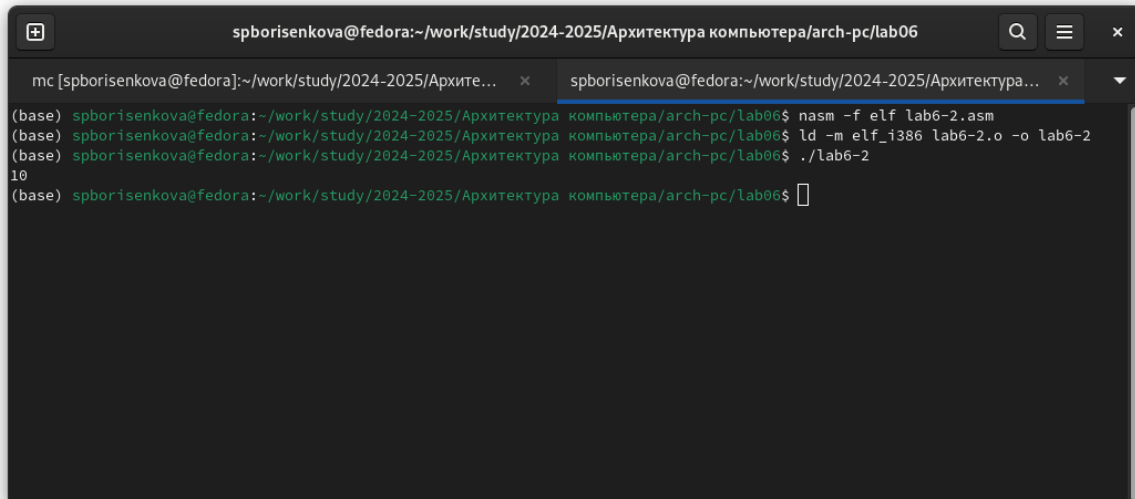


```
GNU nano 7.2 lab6-2.asm Modified
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit

[Cancelled]
```

Рис. 2.10: Изменение файла lab6-2.asm

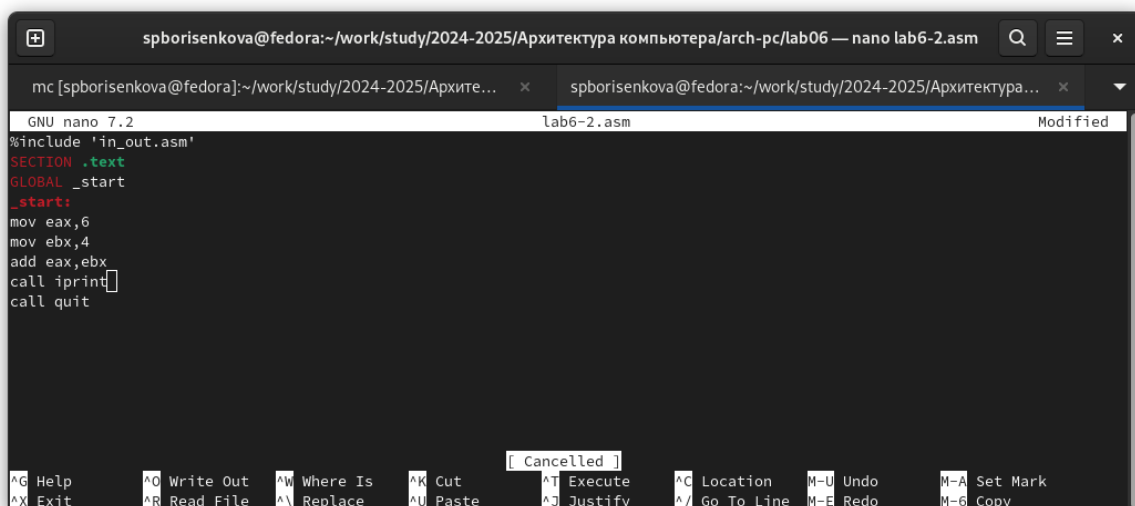
Соберём программу и запустим её (Рис. 2.11):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архите... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 lab6-2.o -o lab6-2
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.11: Сборка исполняемого файла и результат работы программы

Как видим, программа действительно вывела число 10. Кроме операции `iprintLF` в файле `in_out.asm` есть операция `iprint`. Посмотрим, чем они отличаются. Заменим в коде `iprintLF` на `iprint` (Рис. 2.12):

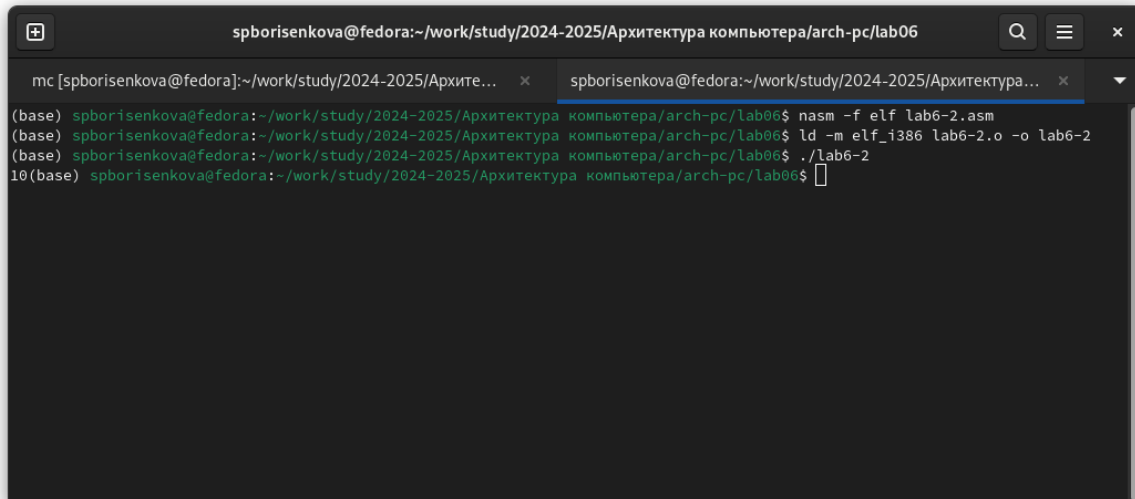


```
GNU nano 7.2 lab6-2.asm Modified
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit

[Cancelled]
```

Рис. 2.12: Редактирование файла lab6-2.asm

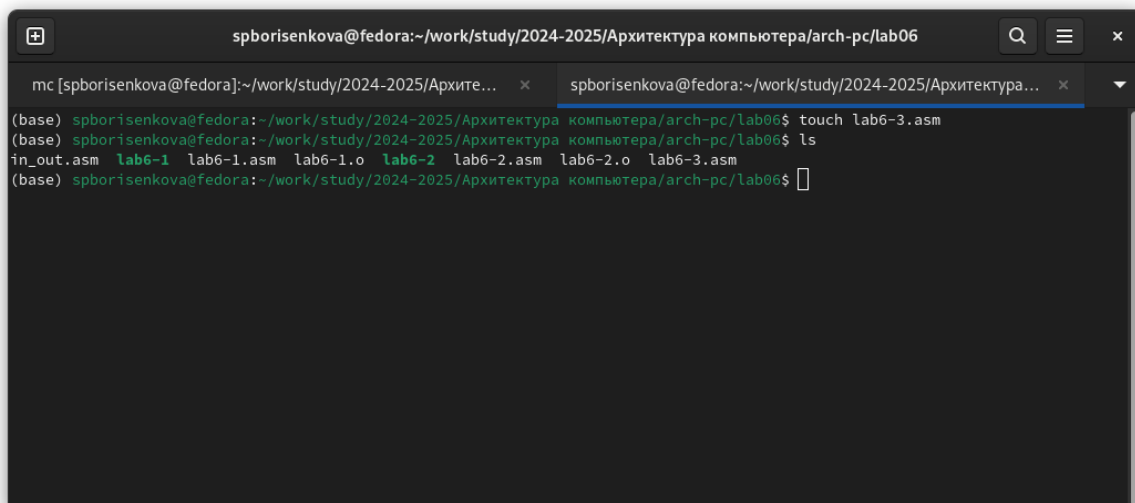
Попробуем собрать программу и запустить её (Рис. 2.13):

A terminal window with a dark background. The title bar shows the user 'spborisenkova@fedora' and the directory '~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06'. The terminal shows the following commands and output:

```
mc [spborisenkova@fedora:~/work/study/2024-2025/Архите... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 lab6-2.o -o lab6-2
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.13: Сборка и результат работы отредактированного файла

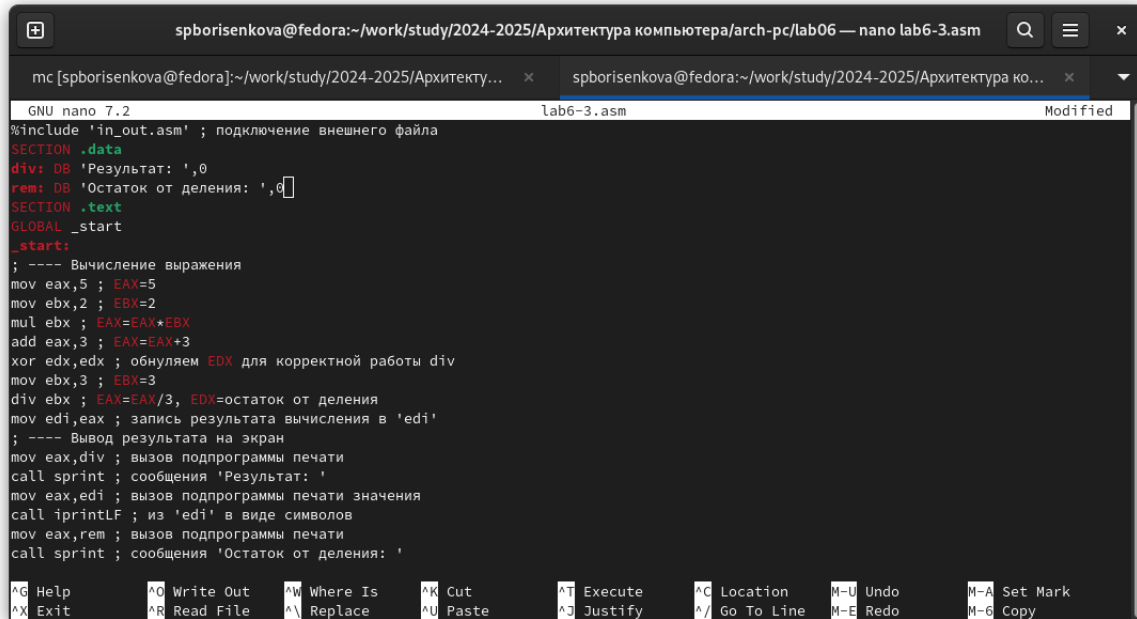
Как видим, операция `iprint` не переносит на следующую строку, в отличие от `iprintLF`. Теперь создадим третий файл (Рис. 2.14):

A terminal window with a dark background. The title bar shows the user 'spborisenkova@fedora' and the directory '~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06'. The terminal shows the following commands and output:

```
mc [spborisenkova@fedora:~/work/study/2024-2025/Архите... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-3.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/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.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.14: Создание файла lab6-3.asm

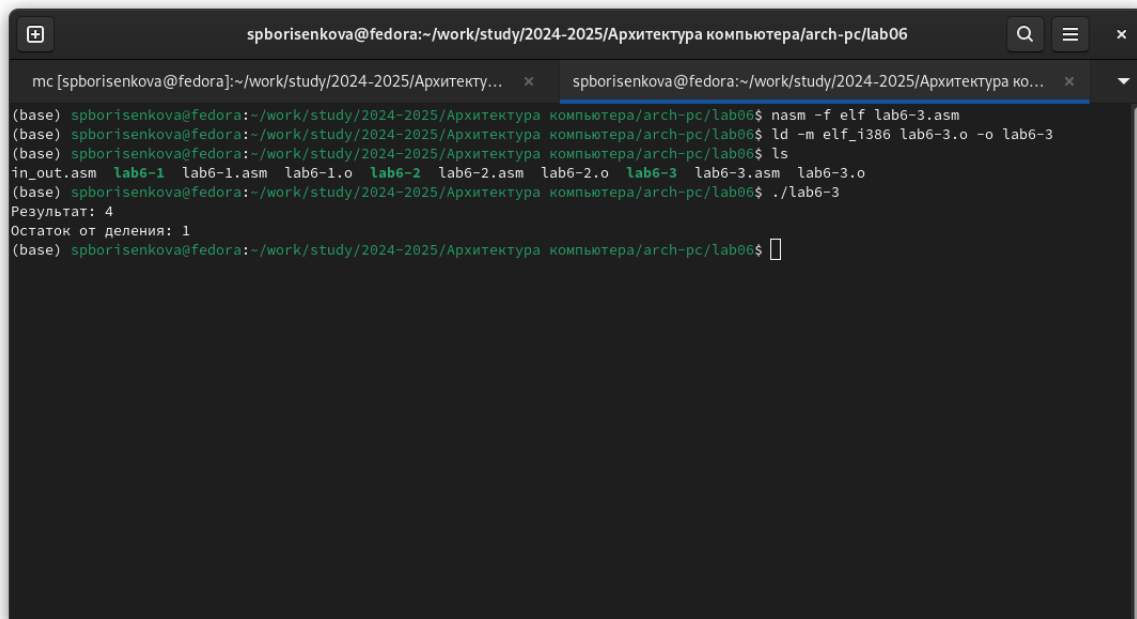
Он должен выводить значение функции  $(5 \cdot 2 + 3) / 3$ . Для этого вставим код из файла листинга 6.3 (Рис. 2.15):



```
GNU nano 7.2 lab6-3.asm Modified
#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: '
```

Рис. 2.15: Вставка кода из листинга в файл

Попробуем запустить эту программу (Рис. 2.16):

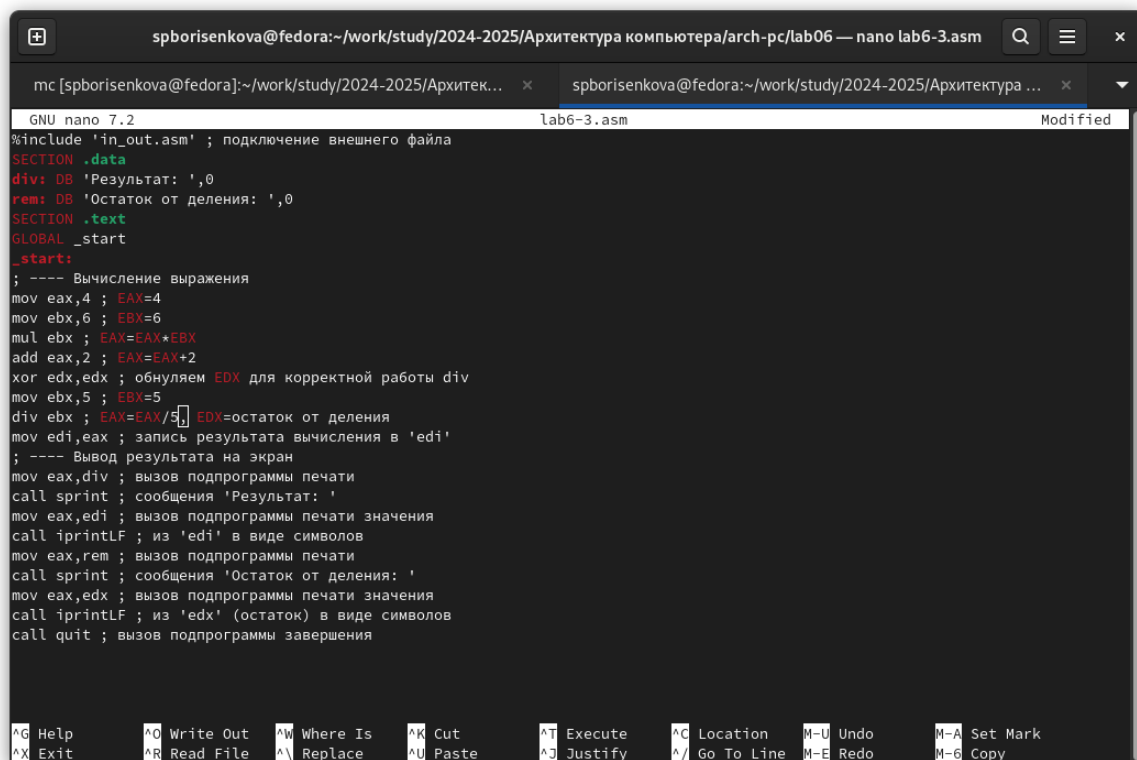


```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архитекту... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура ко... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 lab6-3.o -o lab6-3
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/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
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.16: Сборка файла lab6-3.asm и результат его работы

Полученный результат совпадает с результатом, указанным в лабораторной

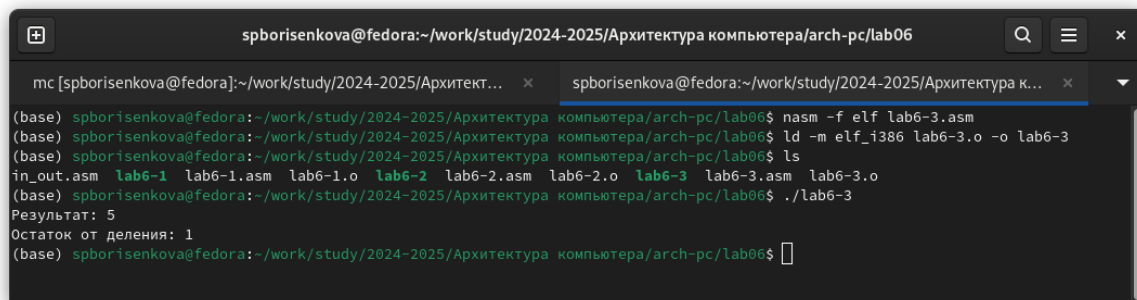
работе. Теперь изменим файл так, чтобы он вычислял значение выражения  $(4*6+2)/5$ . Для этого поменяем числа в коде следующим образом(Рис. 2.17):



```
GNU nano 7.2 lab6-3.asm Modified
#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 iprintf ; 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 iprintf ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 2.17: Редактирование файла lab6-3.asm

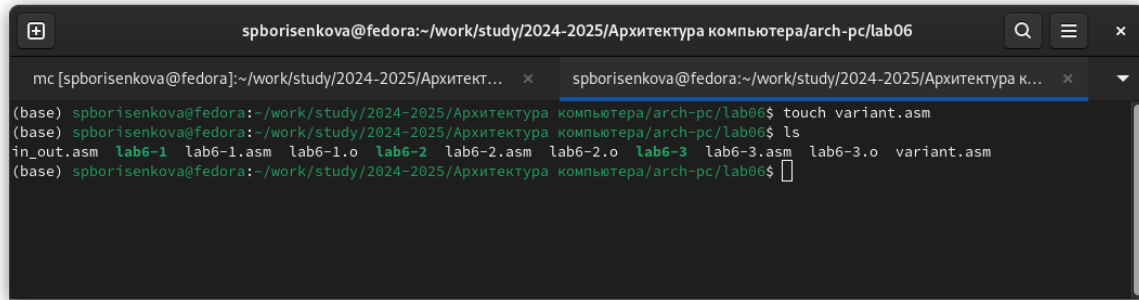
Соберём программу и запустим её (Рис. 2.18):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora:~/work/study/2024-2025/Архитек... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура к... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-3.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 lab6-3.o -o lab6-3
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/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
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.18: Сборка файла lab6-3.asm и результат его работы

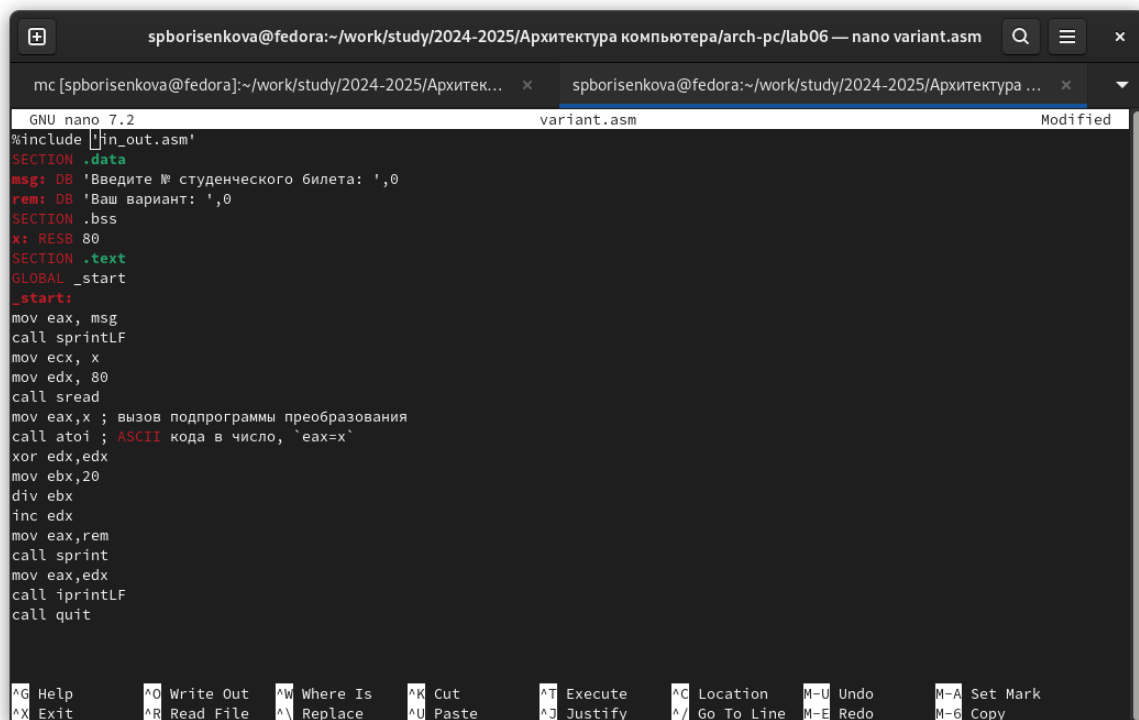
Вывод корректный. Теперь создадим файл variant.asm для вычисления варианта самостоятельной работы (Рис. 2.19):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архитек... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура к... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch variant.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/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
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.19: Создание файла variant.asm для вычисления варианта для самостоятельной работы

Вставим в файл код из листинга 6.4 (Рис. 2.20):

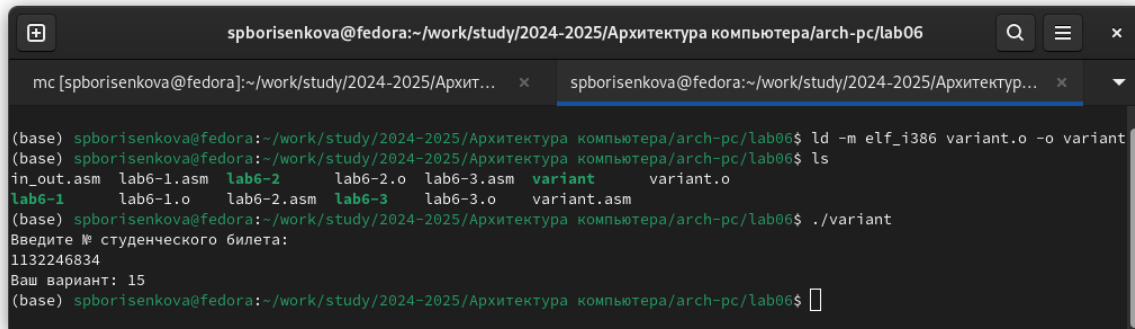


```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06 — nano variant.asm
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архитек... x spborisenkova@fedora:~/work/study/2024-2025/Архитектура ... x
GNU nano 7.2 variant.asm Modified
%include 'in_out.asm'
SECTION .data
msg: 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 read
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 iprintf
call quit

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^_ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo M-G Copy
```

Рис. 2.20: Вставка кода в файл variant.asm

Соберём и запустим программу, указав номер студенческого билета. В моём случае это 1132246834 (Рис. 2.21):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архит... x spborisenkova@fedora:~/work/study/2024-2025/Архитектур... x
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 variant.o -o variant
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ls
in_out.asm lab6-1.asm lab6-2 lab6-2.o lab6-3.asm variant variant.o
lab6-1 lab6-1.o lab6-2.asm lab6-3 lab6-3.o variant.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246834
Ваш вариант: 15
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 2.21: Сборка и запуск программы variant

Мой вариант - 15.

Ответы на вопросы:

1. За это отвечает 25-ая строчка `call sprint`, перед которой идёт строка `mov eax,ret`, которая перемещает строку с фразой в регистр `eax`, из которого мы считываем данные для вывода

2. Эти строки используются для того, чтобы записать данные в переменную `x`

3. Для преобразования ASCII кода в число

4. Напрямую за вычисление отвечают следующие строки:

```
div ebx
```

```
inc edx
```

5. В регистр `edx`

6. Для увеличения значения регистра `edx` на единицу

7. За это отвечают строки:

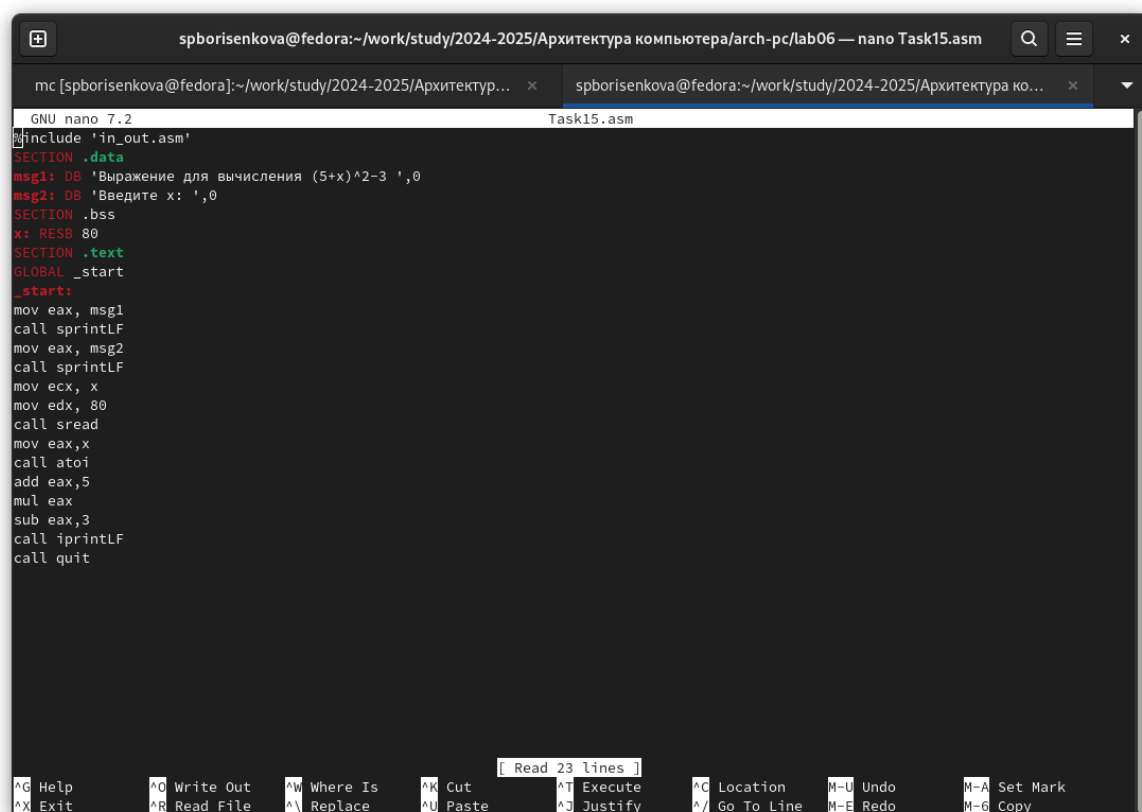
```
mov eax,edx
```

```
call iprintLF
```



### 3 Выполнение задания для самостоятельной работы

Теперь в качестве самостоятельной работы напомним код программы для вычисления выражения в варианте 15:  $(5+x)^2-3$ . В предварительно созданном файле Task15.asm напишем следующий код (Рис. 3.1):

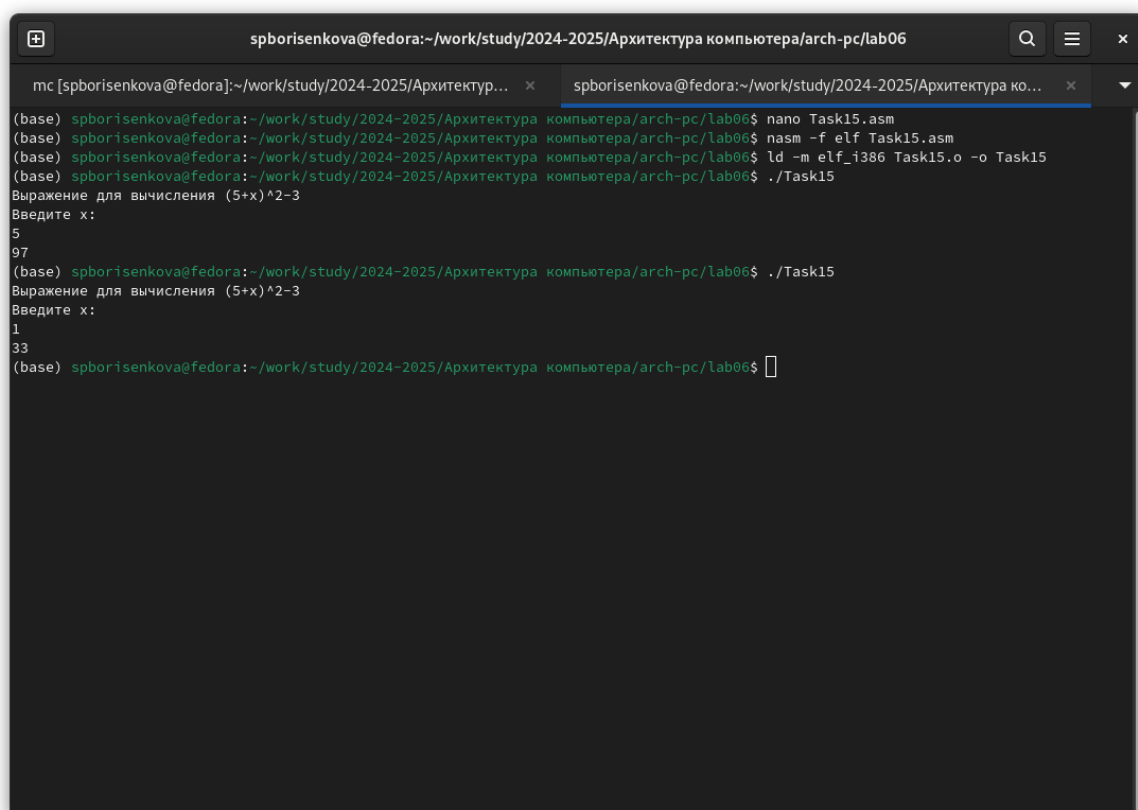


```
GNU nano 7.2 Task15.asm
#include 'in_out.asm'
SECTION .data
msg1: DB 'Выражение для вычисления (5+x)^2-3 ',0
msg2: DB 'Введите x: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg1
call sprintf
mov eax, msg2
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
add eax, 5
mul eax
sub eax, 3
call iprintf
call quit
```

The screenshot shows a terminal window with the nano text editor open to a file named Task15.asm. The code is written in assembly language and includes instructions for printing messages, reading input, and performing arithmetic operations. The terminal window title is 'spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06 — nano Task15.asm'. The bottom of the window displays a list of keyboard shortcuts for the nano editor.

Рис. 3.1: Код требуемой программы

И запустим его, указав в качестве  $x$  предложенные в лабораторной работе значения (Рис. 3.2):



```
spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
mc [spborisenkova@fedora]:~/work/study/2024-2025/Архитектур... × spborisenkova@fedora:~/work/study/2024-2025/Архитектура ко... ×
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nano Task15.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf Task15.asm
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 Task15.o -o Task15
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./Task15
Выражение для вычисления (5+x)^2-3
Введите x:
5
97
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./Task15
Выражение для вычисления (5+x)^2-3
Введите x:
1
33
(base) spborisenkova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.2: Запуск программы и проверка её корректной работы

## **4 Выводы**

В результате выполнения лабораторной работы было получено представление о том, какие арифметические операции есть в языке ассемблера, и как они работают. Были написаны программы, использующие в себе операции сложения, вычитания, умножения и деления.