

# **Отчёт по лабораторной работе 6**

**Архитектура компьютера**

Исаев Кирилл НБИбд-01-24

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Символьные и численные данные в NASM . . . . .	6
2.2	Выполнение арифметических операций в NASM . . . . .	12
2.2.1	Ответы на вопросы по программе variant.asm . . . . .	17
2.3	Выполнение заданий для самостоятельной работы . . . . .	18
<b>3</b>	<b>Выводы</b>	<b>21</b>

## Список иллюстраций

2.1	Программа в файле lab6-1.asm . . . . .	7
2.2	Запуск программы lab6-1.asm . . . . .	7
2.3	Программа в файле lab6-1.asm . . . . .	8
2.4	Запуск программы lab6-1.asm . . . . .	8
2.5	Программа в файле lab6-2.asm . . . . .	9
2.6	Запуск программы lab6-2.asm . . . . .	9
2.7	Программа в файле lab6-2.asm . . . . .	10
2.8	Запуск программы lab6-2.asm . . . . .	11
2.9	Программа в файле lab6-2.asm . . . . .	11
2.10	Запуск программы lab6-2.asm . . . . .	12
2.11	Программа в файле lab6-3.asm . . . . .	13
2.12	Запуск программы lab6-3.asm . . . . .	13
2.13	Программа в файле lab6-3.asm . . . . .	14
2.14	Запуск программы lab6-3.asm . . . . .	15
2.15	Программа в файле variant.asm . . . . .	16
2.16	Запуск программы variant.asm . . . . .	16
2.17	Программа в файле calc.asm . . . . .	19
2.18	Запуск программы calc.asm . . . . .	20

## **Список таблиц**

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

## 2 Выполнение лабораторной работы

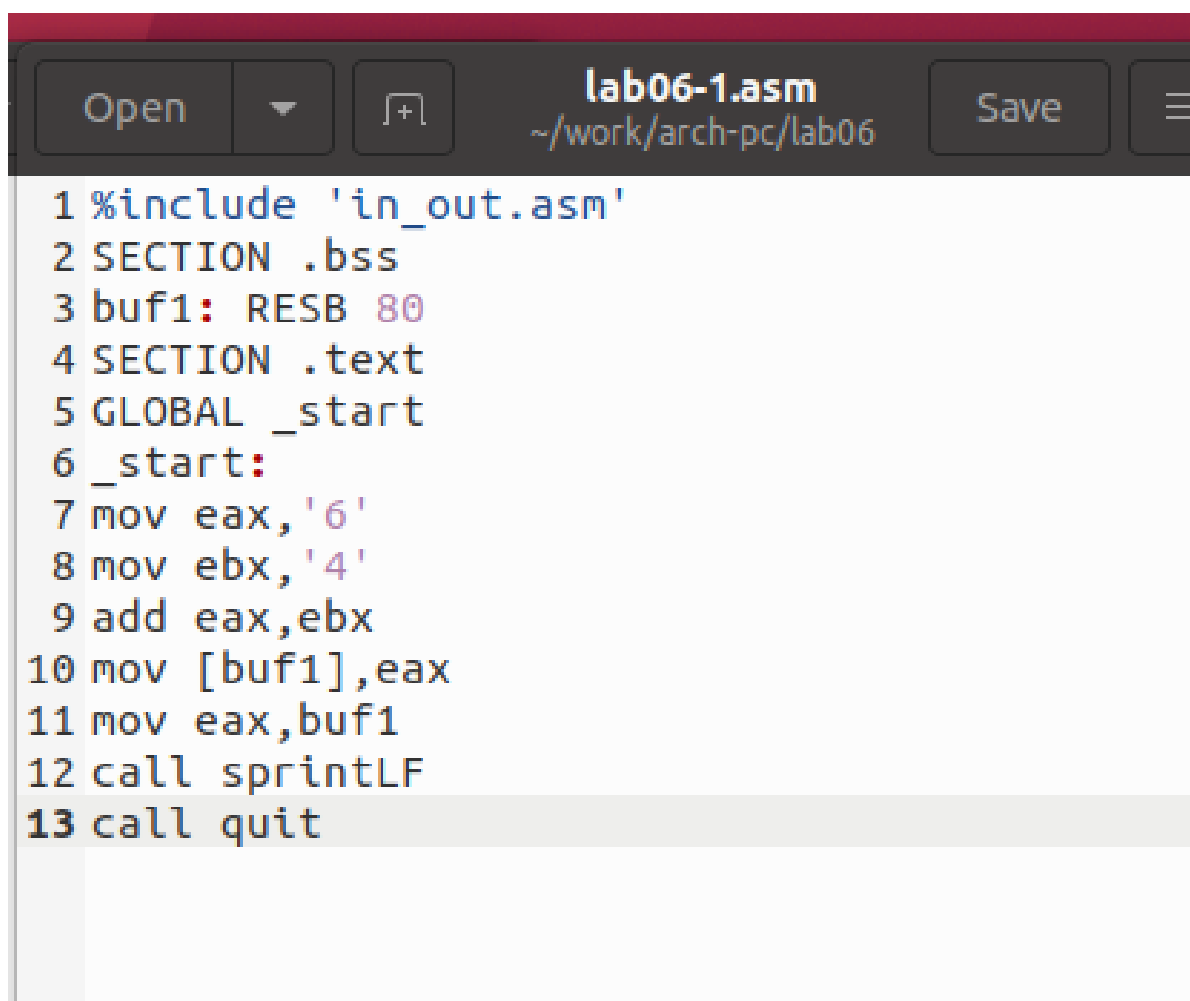
### 2.1 Символьные и численные данные в NASM

Создал каталог для программ лабораторной работы №6, перешел в него и создал файл с названием “lab6-1.asm”.

Рассмотрим примеры программ вывода символьных и численных значений. Программы выводят значения, записанные в регистр `eax`.

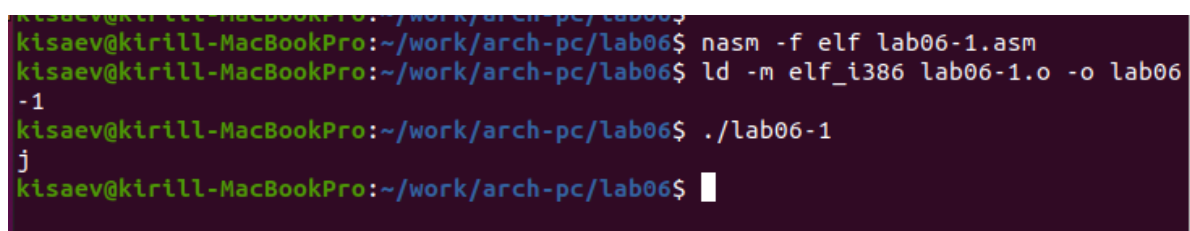
В данной программе в регистр `eax` записан символ '6', а в регистр `ebx` символ '4'. Затем прибавляем значение регистра `ebx` к значению в регистре `eax` (результат сложения записывается в `eax`). После этого выводим результат.

Так как для работы функции `sprintf` в регистр `eax` должен быть записан адрес, используется дополнительная переменная. Значение регистра `eax` записывается в переменную `buf1`, затем адрес переменной `buf1` записывается в регистр `eax`, и вызывается функция `sprintf`.



```
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax, '6'
8 mov ebx, '4'
9 add eax, ebx
10 mov [buf1], eax
11 mov eax, buf1
12 call sprintf
13 call quit
```

Рис. 2.1: Программа в файле lab6-1.asm

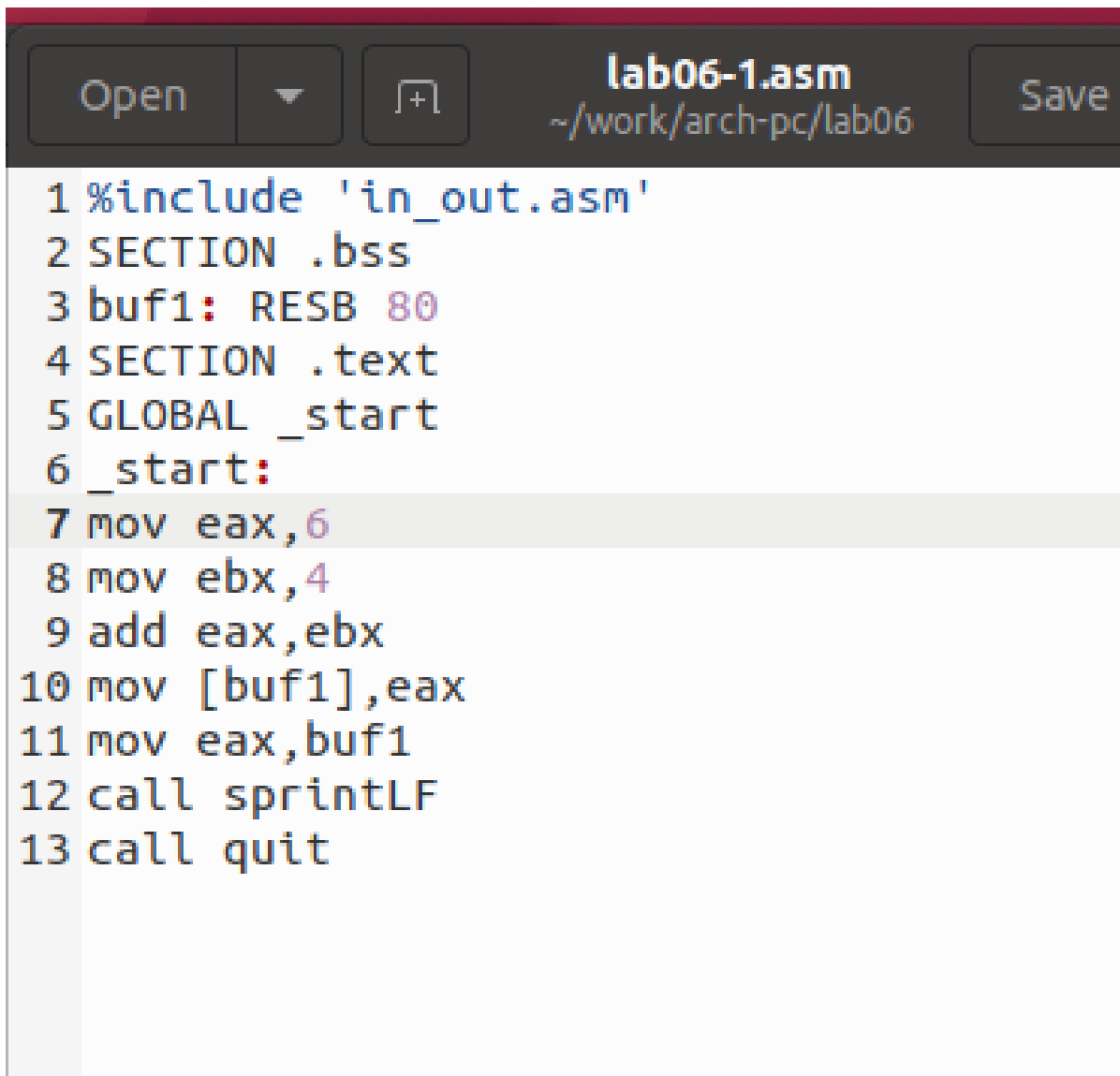


```
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ nasm -f elf lab06-1.asm
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 lab06-1.o -o lab06-1
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./lab06-1
j
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
```

Рис. 2.2: Запуск программы lab6-1.asm

При выводе значения регистра `eax` ожидалось число 10. Однако результатом оказался символ 'j'. Это произошло из-за сложения кодов символов '6' (54 в десятичном представлении) и '4' (52). Команда `add eax, ebx` записала сумму кодов — 106, что является кодом символа 'j'.

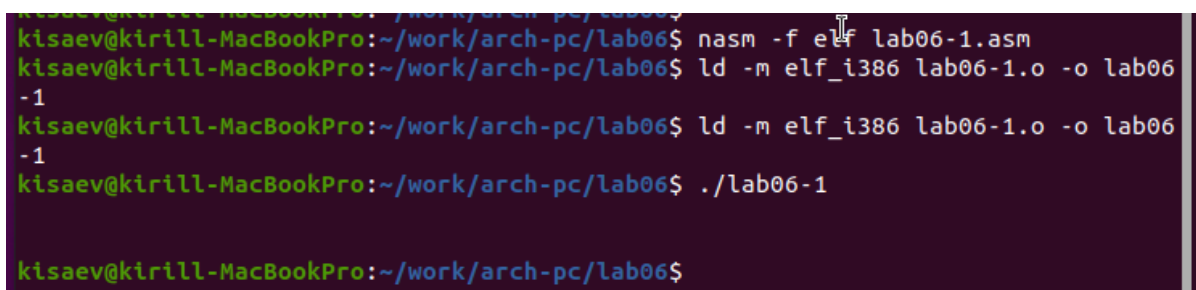
Далее текст программы изменен, вместо символов записаны числа.



```
Open  lab06-1.asm  Save
~/work/arch-pc/lab06

1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,6
8 mov ebx,4
9 add eax,ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintf
13 call quit
```

Рис. 2.3: Программа в файле lab6-1.asm



```
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ nasm -f elf lab06-1.asm
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 lab06-1.o -o lab06-1
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 lab06-1.o -o lab06-1
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./lab06-1

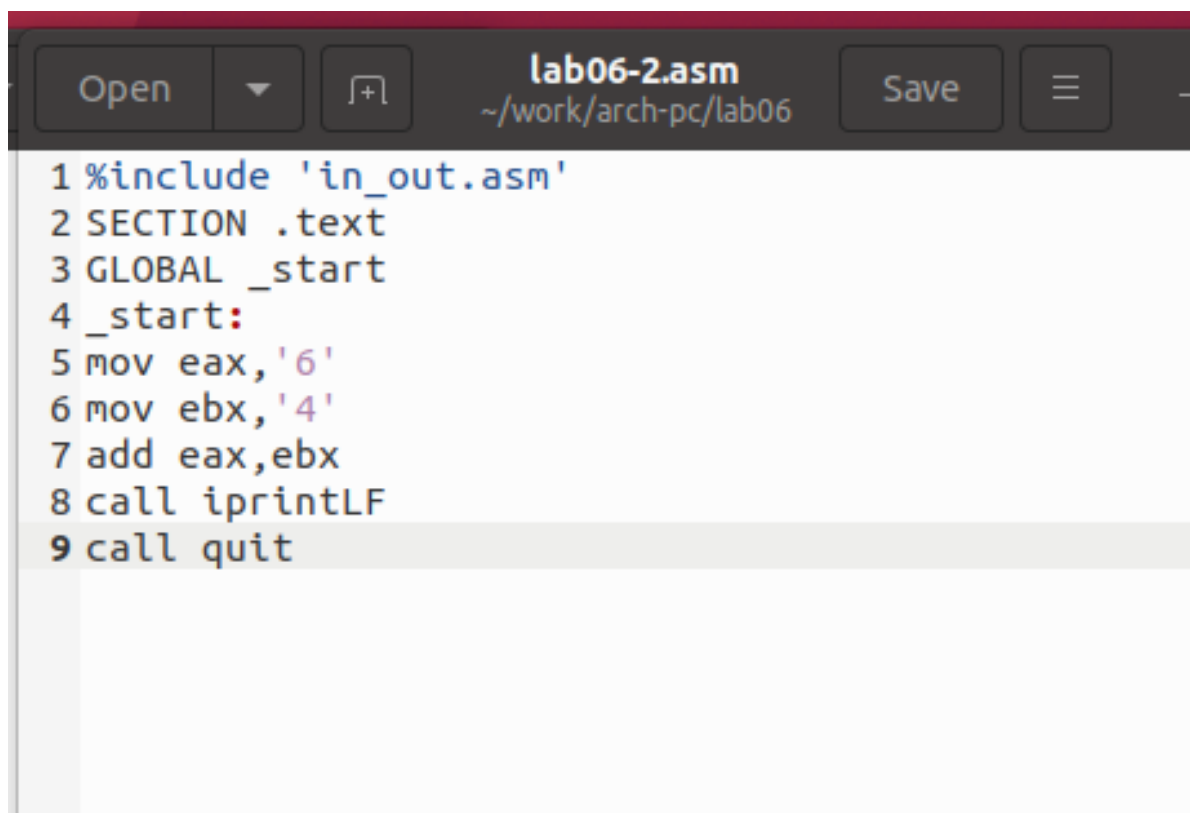
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
```

Рис. 2.4: Запуск программы lab6-1.asm



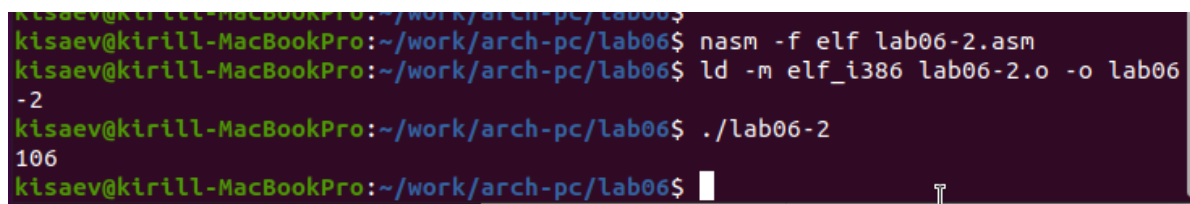
При выполнении программы вместо ожидаемого числа 10 был выведен символ с кодом 10, который соответствует символу возврата каретки (пустая строка в консоли).

В файле `in_out.asm` реализованы подпрограммы для работы с числами и преобразования символов ASCII. Текст программы был модифицирован с использованием этих функций.



```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax, '6'
6 mov ebx, '4'
7 add eax, ebx
8 call iprintLF
9 call quit
```

Рис. 2.5: Программа в файле lab6-2.asm



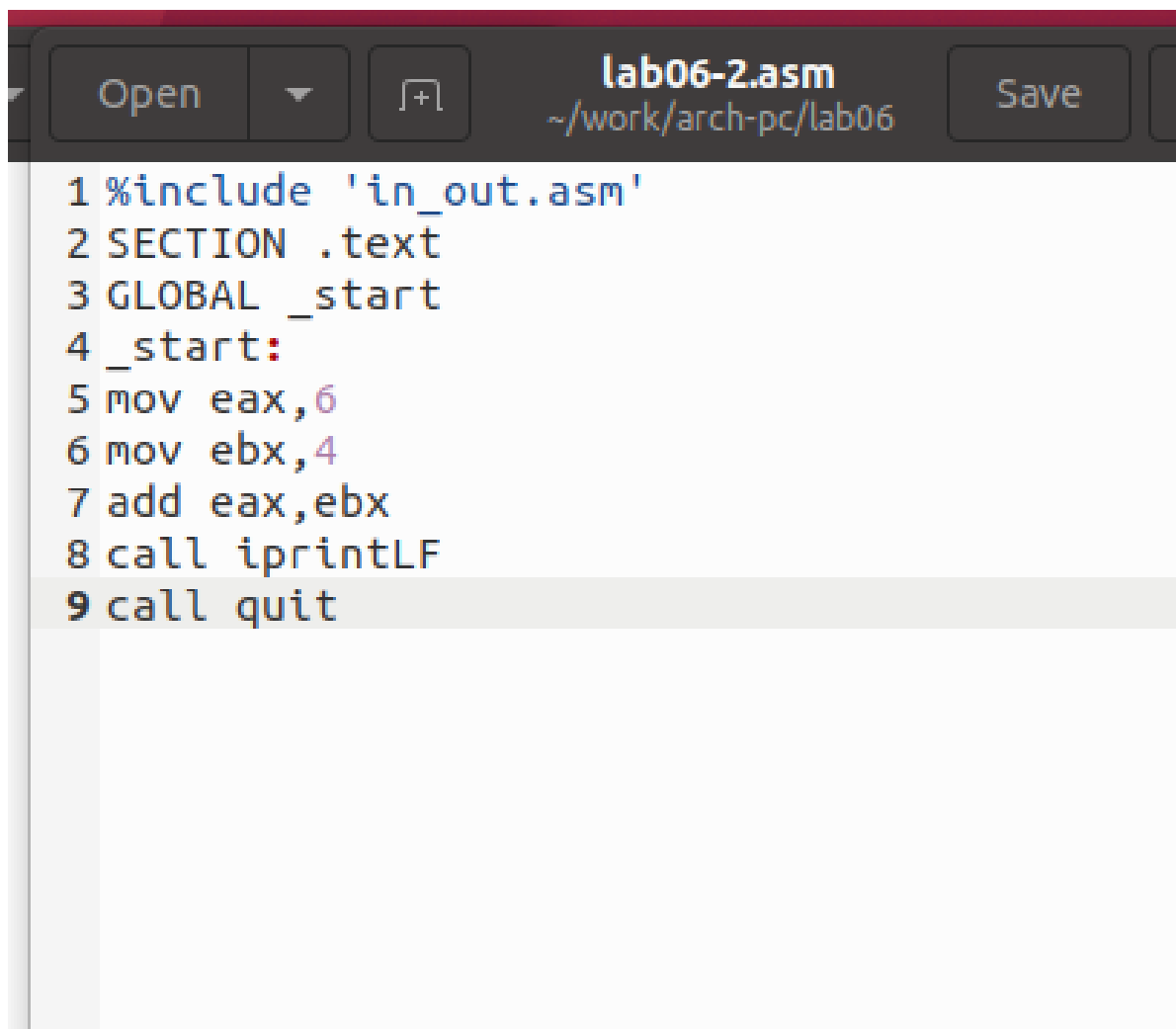
```
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ nasm -f elf lab06-2.asm
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 lab06-2.o -o lab06-2
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./lab06-2
106
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
```

Рис. 2.6: Запуск программы lab6-2.asm

Обновленная программа вывела число 106. Здесь команда `add` складывает

коды символов '6' и '4'. Благодаря функции `iprintLF` было выведено именно число, а не символ с кодом 106.

По аналогии символы заменены на числа.



```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit
```

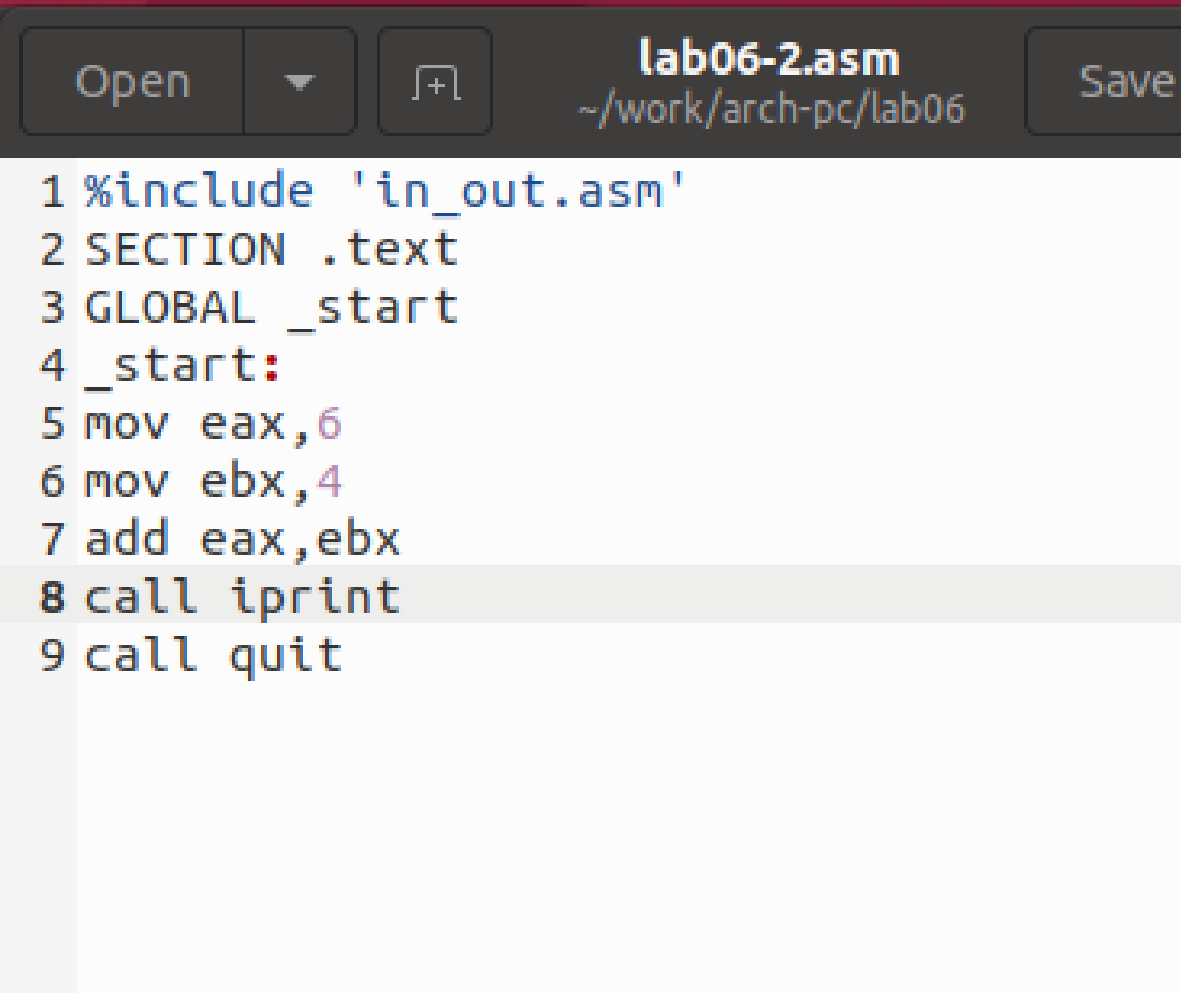
Рис. 2.7: Программа в файле lab6-2.asm

Функция `iprintLF` позволила вывести число 10. На этот раз операндами выступали непосредственно числа, а не коды символов.

```
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ nasm -f elf lab06-2.asm  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 lab06-2.o -o lab06-2  
-2  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./lab06-2  
10  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
```

Рис. 2.8: Запуск программы lab6-2.asm

Функция `iprintLF` была заменена на `iprint`, создан исполняемый файл и запущен. Вывод теперь отличается отсутствием перехода на новую строку.



```
1 %include 'in_out.asm'  
2 SECTION .text  
3 GLOBAL _start  
4 _start:  
5 mov eax,6  
6 mov ebx,4  
7 add eax,ebx  
8 call iprint  
9 call quit
```

Рис. 2.9: Программа в файле lab6-2.asm

```
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ nasm -f elf lab06-2.asm
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 lab06-2.o -o lab06-2
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./lab06-2
10kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
```

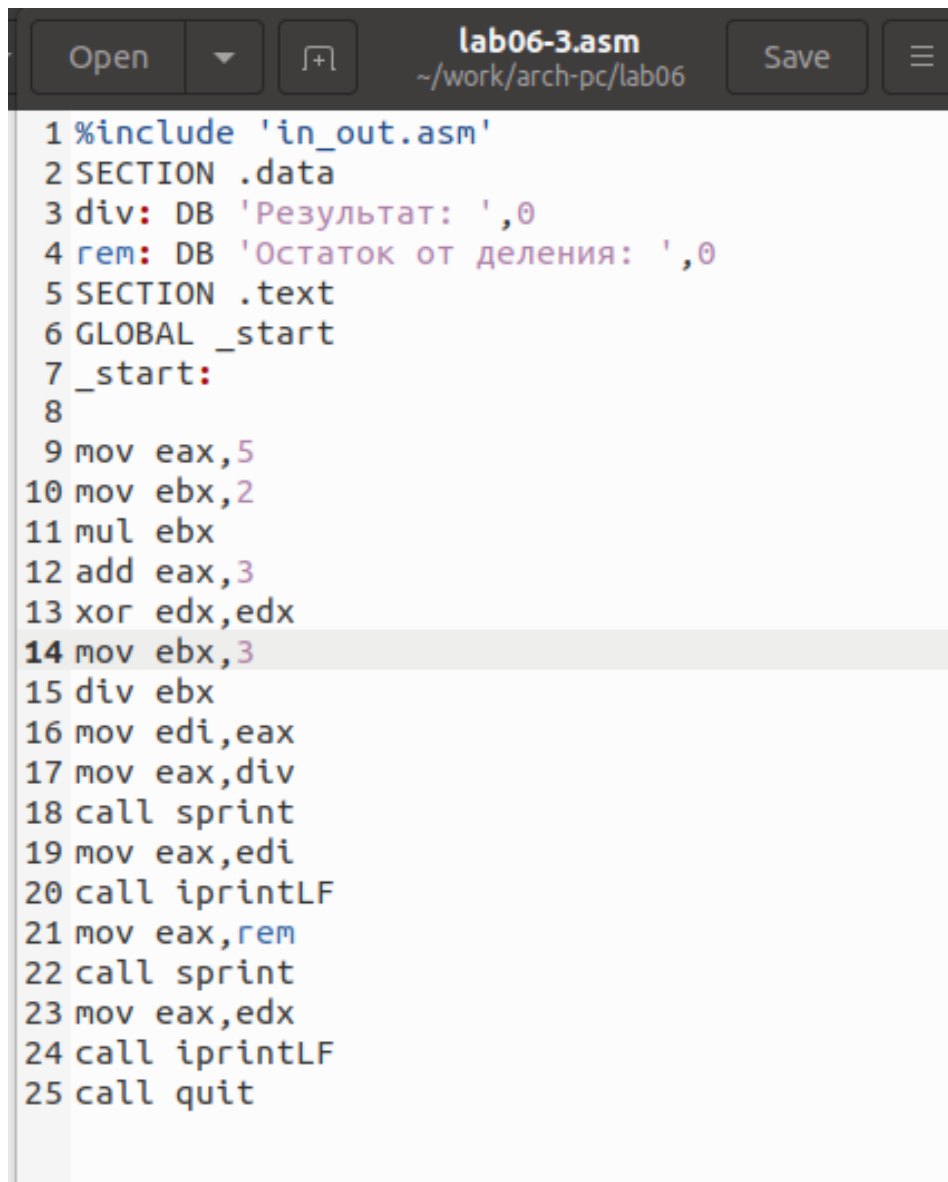
Рис. 2.10: Запуск программы lab6-2.asm

## 2.2 Выполнение арифметических операций в NASM

Пример: вычисление выражения

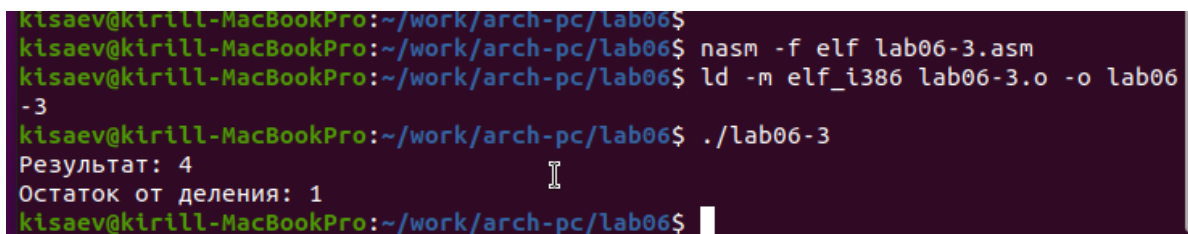
$$f(x) = (5 * 2 + 3) / 3$$

.



```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8
9 mov eax,5
10 mov ebx,2
11 mul ebx
12 add eax,3
13 xor edx,edx
14 mov ebx,3
15 div ebx
16 mov edi,eax
17 mov eax,div
18 call sprint
19 mov eax,edi
20 call iprintLF
21 mov eax,rem
22 call sprint
23 mov eax,edx
24 call iprintLF
25 call quit
```

Рис. 2.11: Программа в файле lab6-3.asm



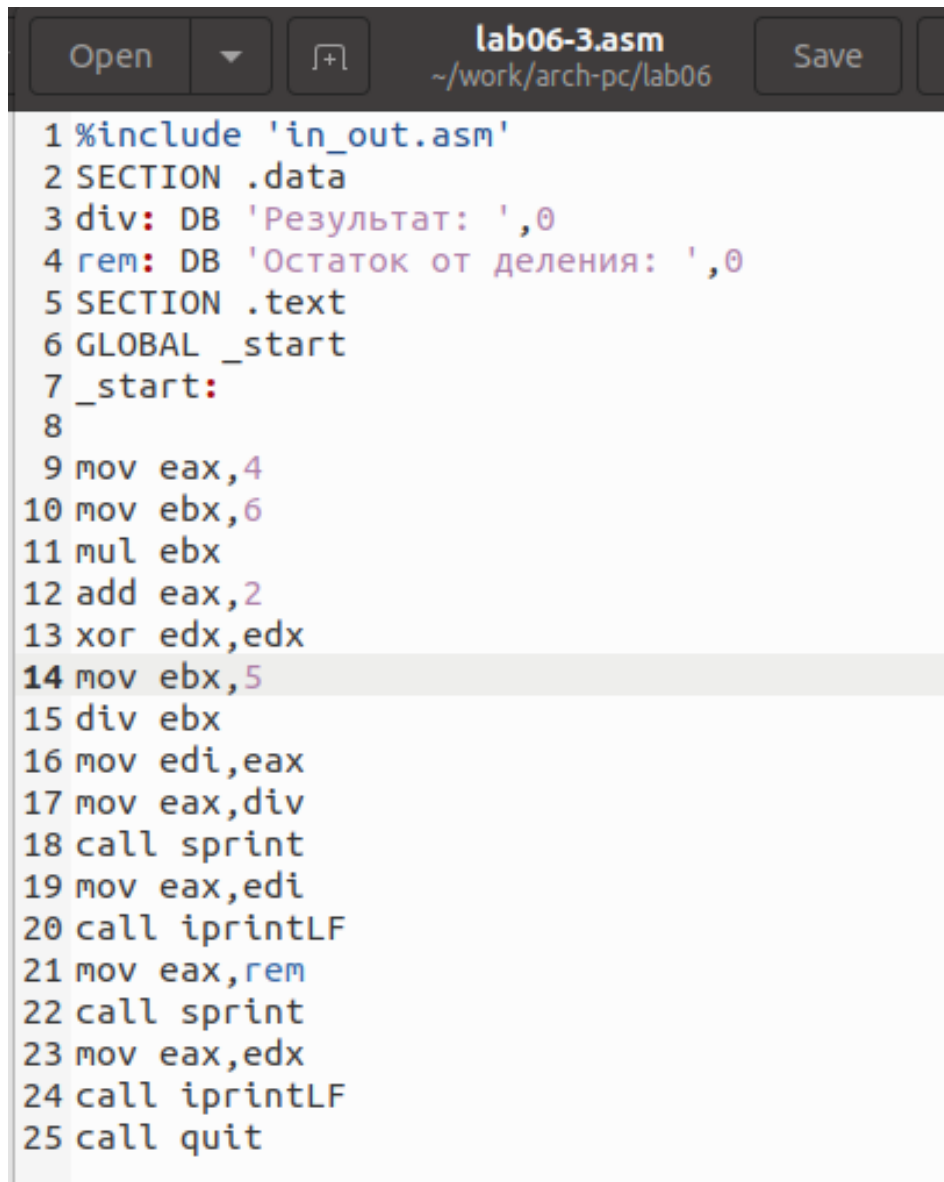
```
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ nasm -f elf lab06-3.asm
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 lab06-3.o -o lab06-3
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./lab06-3
Результат: 4
Остаток от деления: 1
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$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

Рис. 2.12: Запуск программы lab6-3.asm

Изменен текст программы для выражения

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

, создан исполняемый файл, проведена проверка.



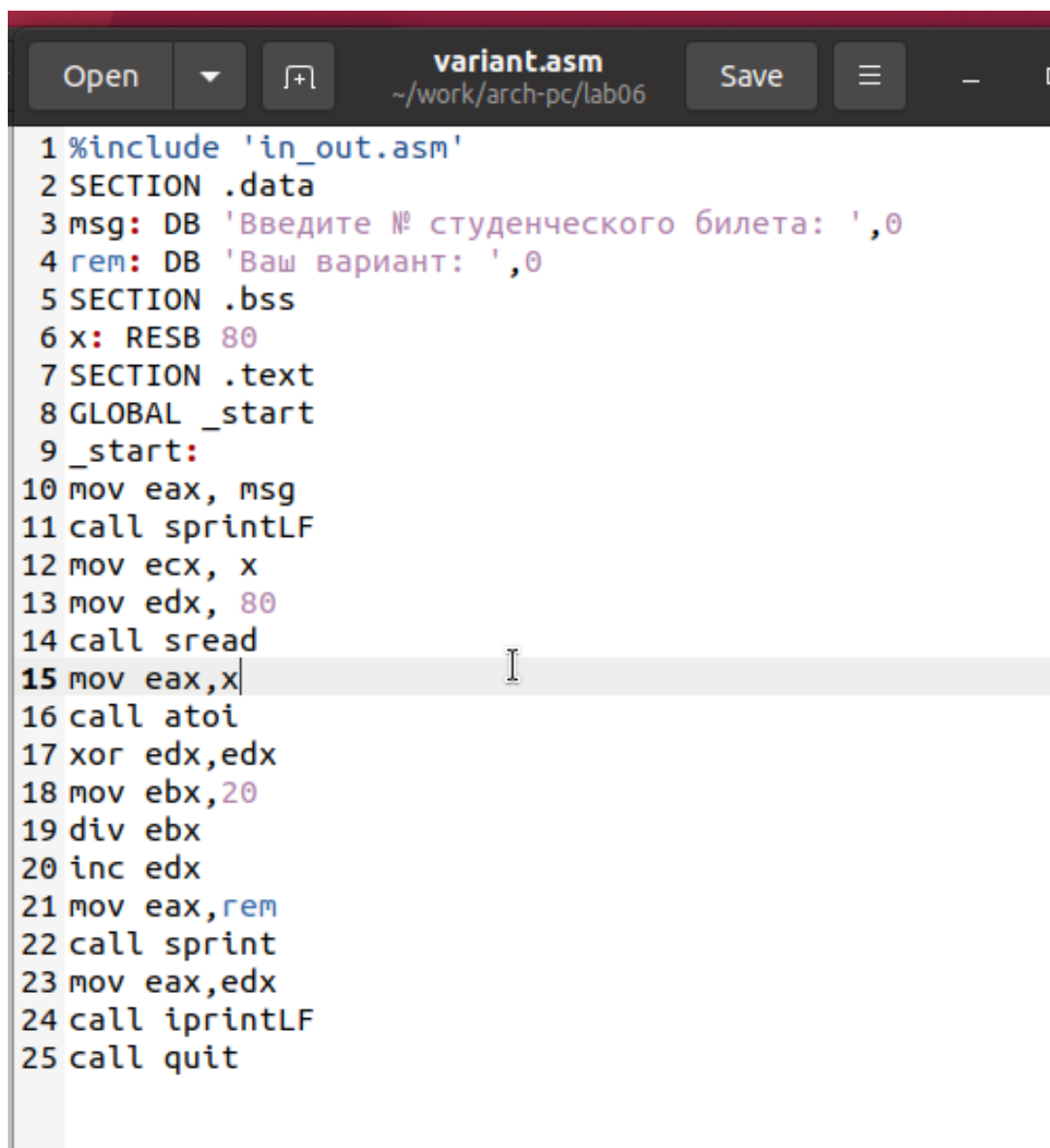
```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8
9 mov eax,4
10 mov ebx,6
11 mul ebx
12 add eax,2
13 xor edx,edx
14 mov ebx,5
15 div ebx
16 mov edi,eax
17 mov eax,div
18 call sprint
19 mov eax,edi
20 call iprintLF
21 mov eax,rem
22 call sprint
23 mov eax,edx
24 call iprintLF
25 call quit
```

Рис. 2.13: Программа в файле lab6-3.asm

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

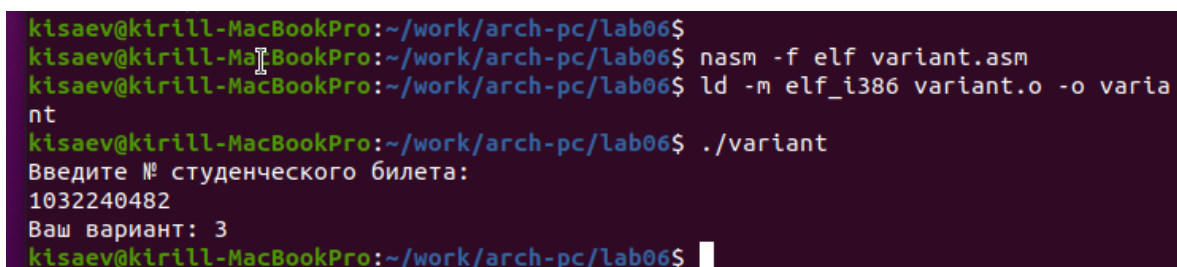
Рис. 2.14: Запуск программы lab6-3.asm

Еще один пример: вычисление варианта задания на основе номера студенческого билета. Число вводится с клавиатуры и преобразуется из символов в число с помощью функции `atoi`.



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите № студенческого билета: ',0
4 rem: DB 'Ваш вариант: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintLF
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x
16 call atoi
17 xor edx, edx
18 mov ebx, 20
19 div ebx
20 inc edx
21 mov eax, rem
22 call sprint
23 mov eax, edx
24 call iprintLF
25 call quit
```

Рис. 2.15: Программа в файле variant.asm



```
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ nasm -f elf variant.asm
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 variant.o -o variant
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032240482
Ваш вариант: 3
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
```

Рис. 2.16: Запуск программы variant.asm



### 2.2.1 Ответы на вопросы по программе variant.asm

1. Какие строки листинга отвечают за вывод сообщения “Ваш вариант:”?

Строки с инструкцией `mov eax, rem` и вызовом `call sprint`.

2. Для чего используется `mov ecx, x`?

Для сохранения значения переменной `x` в регистре `ecx`.

3. Для чего используется `call atoi`?

Для преобразования введенных символов в числовой формат.

4. Какие строки отвечают за вычисление варианта?

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

5. Где записывается остаток от деления при выполнении `div ebx`?

Остаток записывается в регистр `edx`.

6. Для чего используется `inc edx`?

Для увеличения значения `edx` на 1, что соответствует формуле вычисления варианта.

7. Какие строки отвечают за вывод результата вычислений?

- `mov eax, edx`
- `call iprintLF`

## 2.3 Выполнение заданий для самостоятельной работы

Написал программу для вычисления выражения

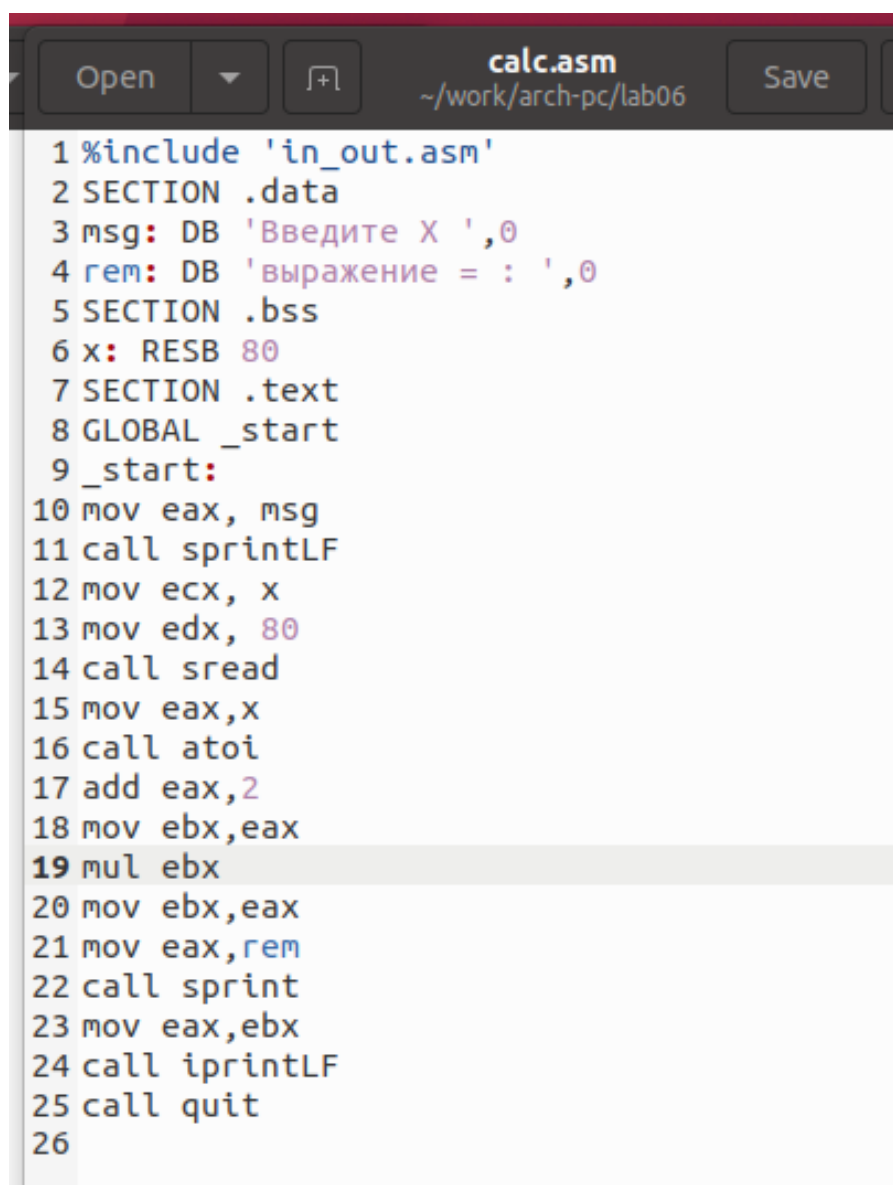
$$y = f(x)$$

. Программа запрашивает значение  $x$ , вычисляет выражение и выводит результат.

Использован вариант 3:

$$(2 + x)^2$$

для  $x = 2$  и  $x = 8$ .



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите X ',0
4 rem: DB 'выражение = : ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintf
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x
16 call atoi
17 add eax, 2
18 mov ebx, eax
19 mul ebx
20 mov ebx, eax
21 mov eax, rem
22 call sprintf
23 mov eax, ebx
24 call iprintLF
25 call quit
26
```

Рис. 2.17: Программа в файле calc.asm

```
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ nasm -f elf calc.asm  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ld -m elf_i386 calc.o -o calc  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./calc  
Введите X  
2  
выражение = : 16  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$ ./calc  
Введите X  
8  
выражение = : 100  
kisaev@kirill-MacBookPro:~/work/arch-pc/lab06$
```

Рис. 2.18: Запуск программы calc.asm

Программа корректно выполняет расчет.

## **3 Выводы**

Изучил работу с арифметическими операциями в NASM.