

# **Отчёта по лабораторной работе №6**

**Арифметические операции в NASM.**

Семенов Сергей Алексеевич

# Содержание

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Цель работы</b>                                  | <b>4</b>  |
| <b>2</b> | <b>Задание</b>                                      | <b>5</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>               | <b>6</b>  |
| 3.1      | Символьные и численные данные в NASM . . . . .      | 6         |
| 3.2      | Выполнение арифметических операций в NASM . . . . . | 11        |
| 3.3      | Ответы на вопросы по программе . . . . .            | 13        |
| 3.4      | Задание для самостоятельной работы . . . . .        | 14        |
| <b>4</b> | <b>Выводы</b>                                       | <b>16</b> |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 3.1  | Создаем каталог с помощью команды <code>mkdir</code> и файл с помощью команды <code>touch</code> . . . . . | 6  |
| 3.2  | Заполняем файл . . . . .   | 7  |
| 3.3  | Запускаем файл и смотрим на его работу . . . . .   | 7  |
| 3.4  | Изменяем файл . . . . .  | 8  |
| 3.5  | Запускаем файл и смотрим на его работу . . . . .   | 8  |
| 3.6  | Создаем файл . . . . .   | 9  |
| 3.7  | Заполняем файл . . . . .   | 9  |
| 3.8  | Смотрим на работу программы . . . . .  | 10 |
| 3.9  | Изменяем файл . . . . .  | 10 |
| 3.10 | Смотрим на работу программы . . . . .  | 10 |
| 3.11 | Заполняем файл . . . . .   | 11 |
| 3.12 | Смотрим на результат работы программы . . . . .  | 12 |
| 3.13 | Смотрим на результат работы программы . . . . .  | 12 |
| 3.14 | Проверяемс результат работы программы . . . . .  | 13 |
| 3.15 | Создаем файл . . . . .   | 14 |
| 3.16 | Проверяем работу программы . . . . .   | 15 |

# 1 Цель работы

Освоить арифметических инструкций языка ассемблера NASM и написать программы для вычисления арифметических выражений с неизвестной.

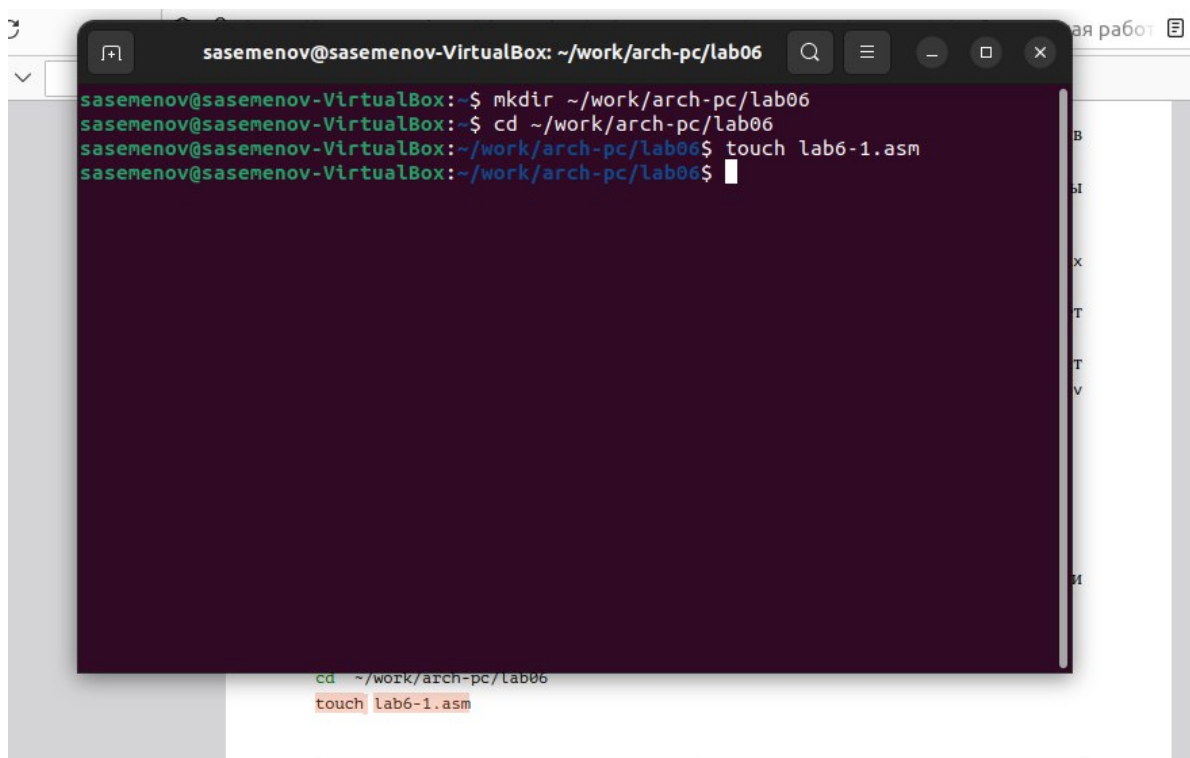
## 2 Задание

Написать программы для решения выражений.

## 3 Выполнение лабораторной работы

### 3.1 Символьные и численные данные в NASM

Создаем каталог для программ ЛБ6, и в нем создаем файл

A screenshot of a terminal window titled 'sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab06'. The terminal shows the following commands and their outputs: 'mkdir ~/work/arch-pc/lab06', 'cd ~/work/arch-pc/lab06', and 'touch lab6-1.asm'. The prompt is '~/.work/arch-pc/lab06\$'.

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

Рис. 3.1: Создаем каталог с помощью команды `mkdir` и файл с помощью команды `touch`

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1

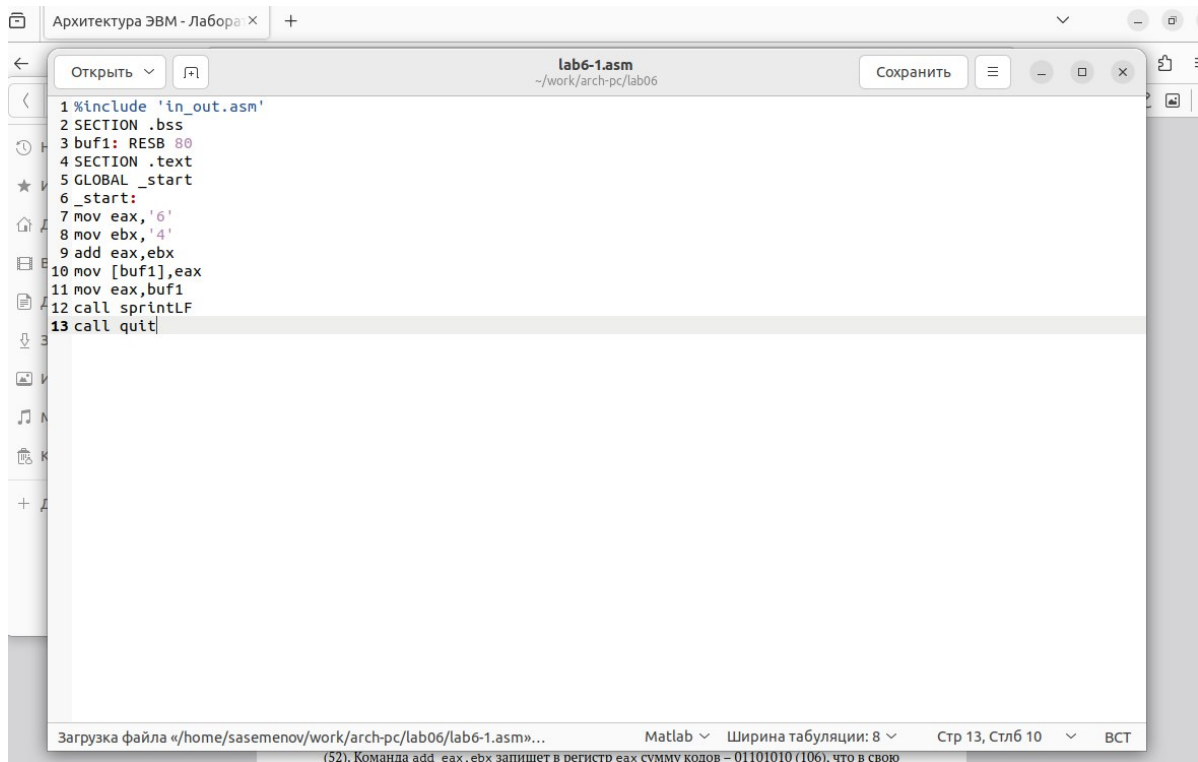


Рис. 3.2: Заполняем файл

Создаем исполняемый файл и запускаем его.

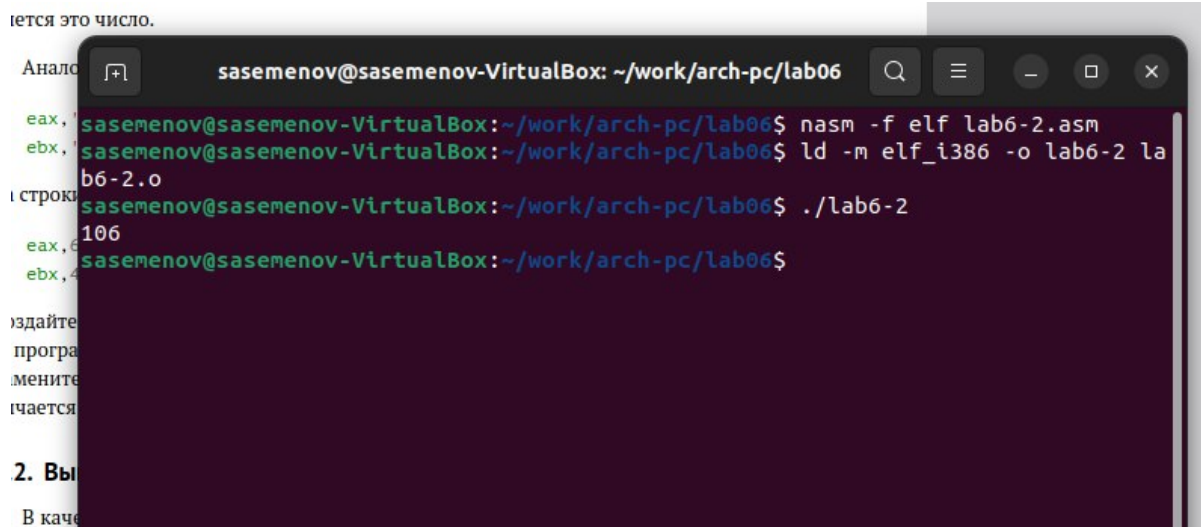


Рис. 3.3: Запускаем файл и смотрим на его работу

Снова открываем файл для редактирования и убираем кавычки с числовых значений

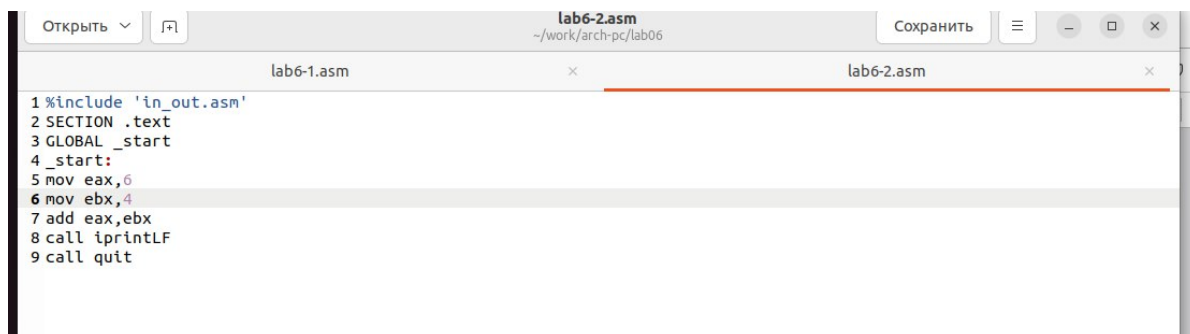


Рис. 3.4: Изменяем файл

Создаем исполняемый файл и запускаем его

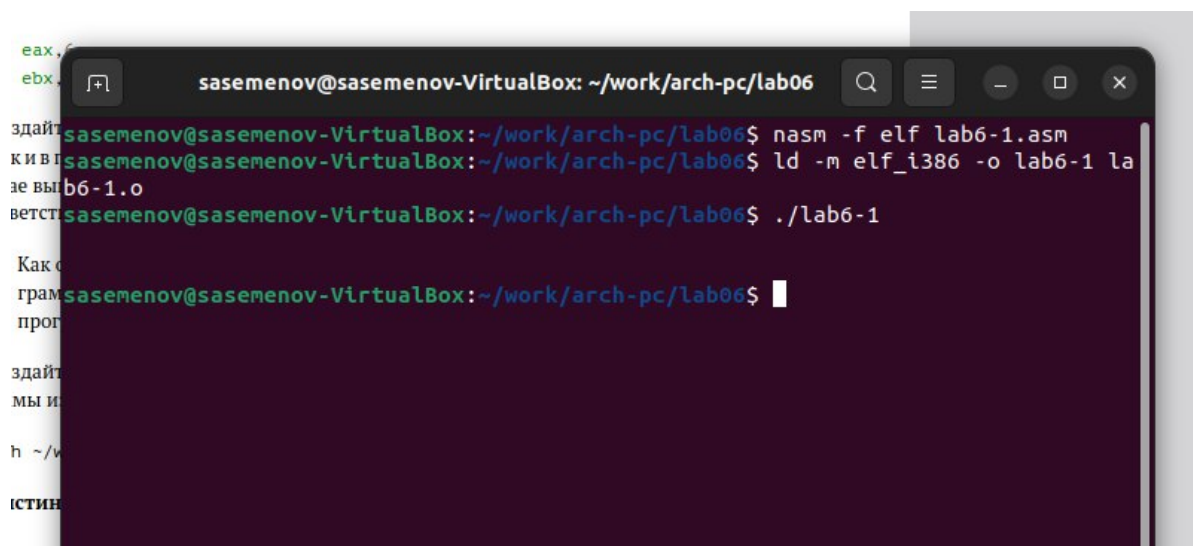


Рис. 3.5: Запускаем файл и смотрим на его работу

Создаем новый файл в каталоге



```
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/  
lab6-2.asm  
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 3.6: Создаем файл

Заполняем файл в соответствии с листингом 6.2

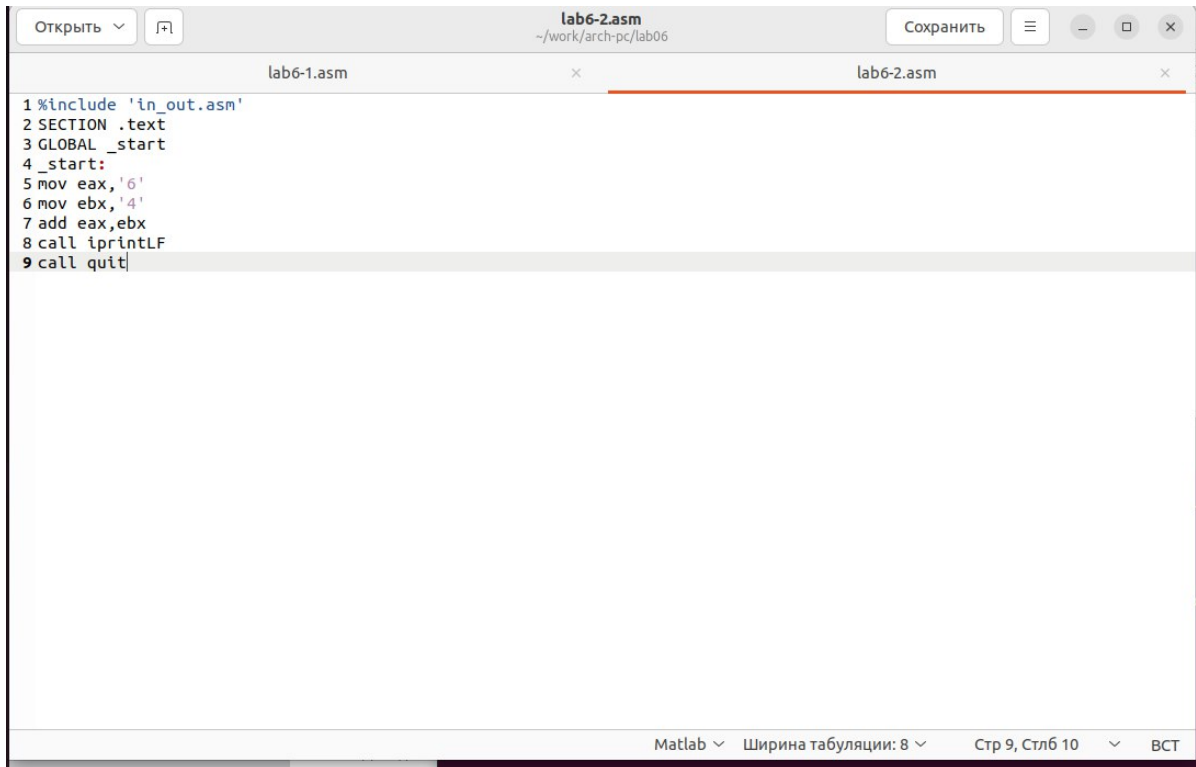


Рис. 3.7: Заполняем файл

Создаем исполняемый файл и запускаем его

ется это число.

Анало

```
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab06
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 la
b6-2.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
106
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$
```

здайте

програ

мените

чается

## 2. Вы

В кач

Рис. 3.8: Смотрим на работу программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений

```
Открыть  lab6-2.asm  Сохранить
~/work/arch-pc/lab06
lab6-1.asm  lab6-2.asm
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. 3.9: Изменяем файл

Создаем исполняемый файл и запускаем его

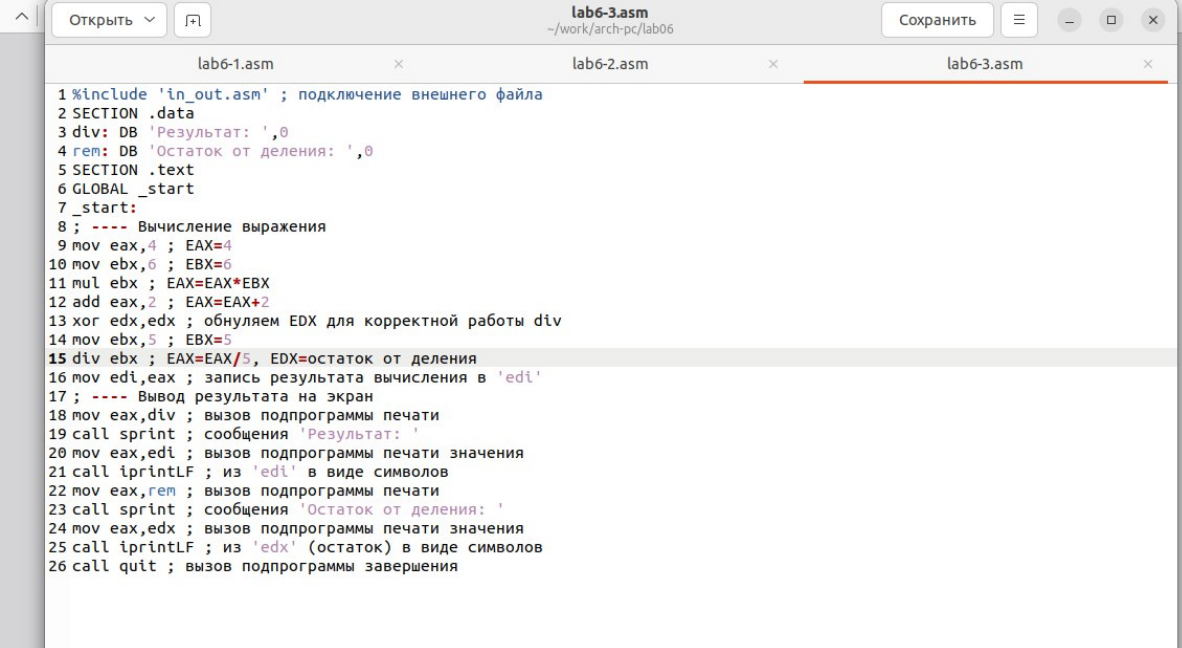
```
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 la
b6-2.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 3.10: Смотрим на работу программы

Вывод функций `iprintLF` и `iprint` отличаются только тем, что `LF` переносит на новую строку.

## 3.2 Выполнение арифметических операций в NASM

Открываем файл и редактируем в соответствии с листингом 6.3



```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ---- Вычисление выражения
9 mov eax,4 ; EAX=4
10 mov ebx,6 ; EBX=6
11 mul ebx ; EAX=EAX*EBX
12 add eax,2 ; EAX=EAX+2
13 xor edx,edx ; обнуляем EDX для корректной работы div
14 mov ebx,5 ; EBX=5
15 div ebx ; EAX=EAX/5, EDX=остаток от деления
16 mov edi,eax ; запись результата вычисления в 'edi'
17 ; ---- Вывод результата на экран
18 mov eax,div ; вызов подпрограммы печати
19 call sprint ; сообщения 'Результат: '
20 mov eax,edi ; вызов подпрограммы печати значения
21 call iprintLF ; из 'edi' в виде символов
22 mov eax,rem ; вызов подпрограммы печати
23 call sprint ; сообщения 'Остаток от деления: '
24 mov eax,edx ; вызов подпрограммы печати значения
25 call iprintLF ; из 'edx' (остаток) в виде символов
26 call quit ; вызов подпрограммы завершения
```

Рис. 3.11: Заполняем файл

Создаем исполняемый файл и запускаем

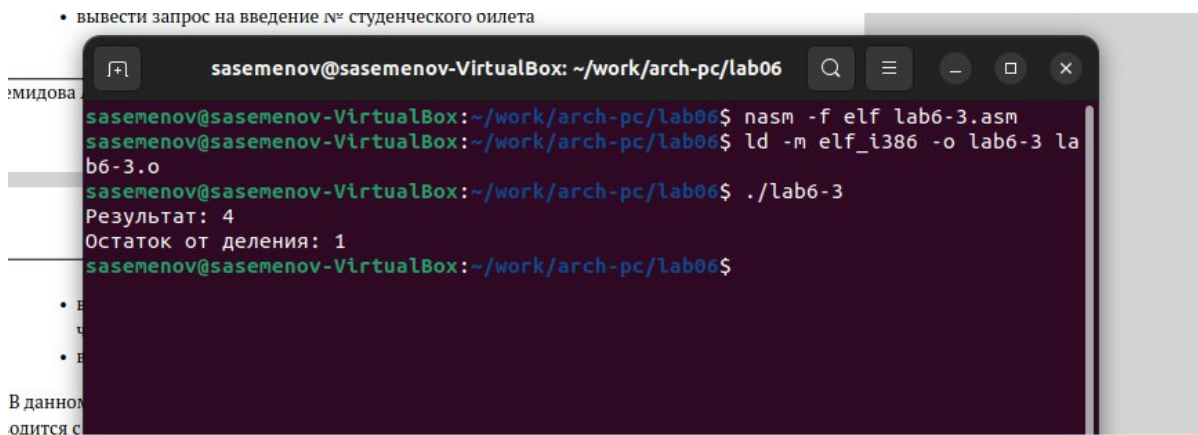


Рис. 3.12: Смотрим на результат работы программы

### Компилируем файл и запускаем программу

ните текст программы для вычисления выражения  $f(x) = (4 * 6 + 2)/5$ . Создайте темный файл и проверьте его работу.

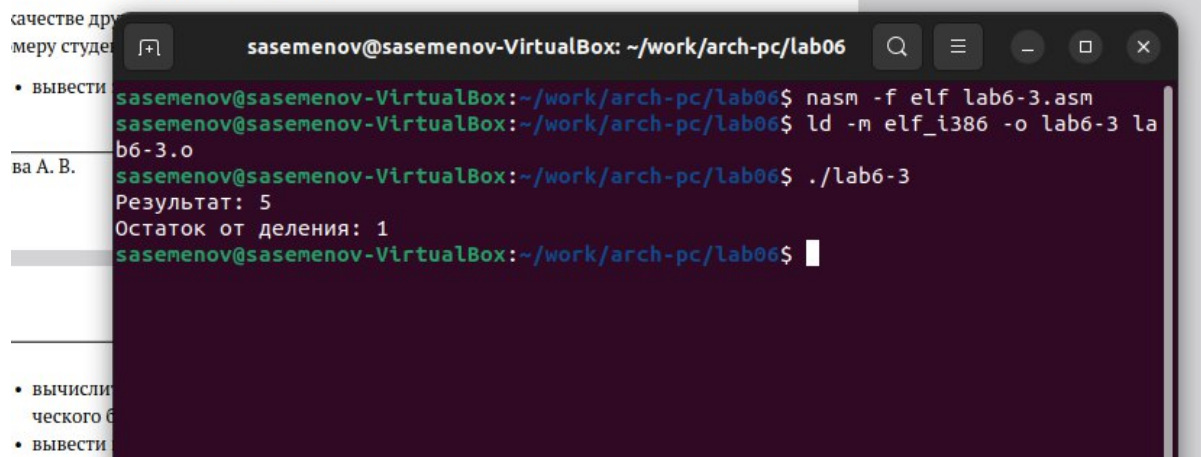
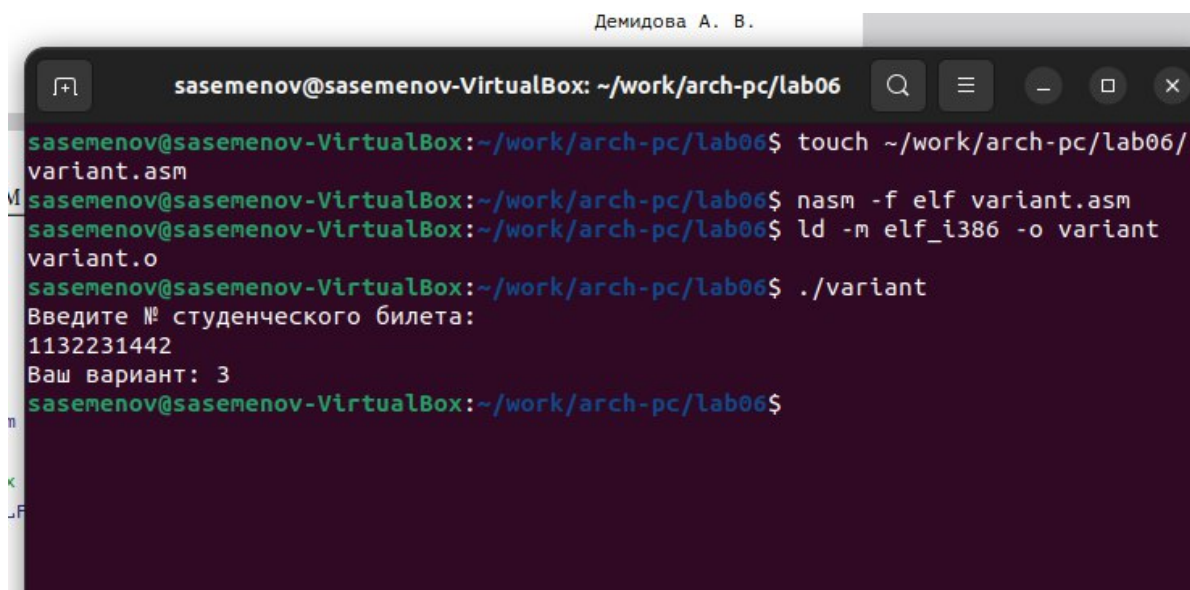


Рис. 3.13: Смотрим на результат работы программы

### Компилируем файл и запускаем его



```
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab06
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf variant.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132231442
Ваш вариант: 3
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 3.14: Проверяем результат работы программы

### 3.3 Ответы на вопросы по программе

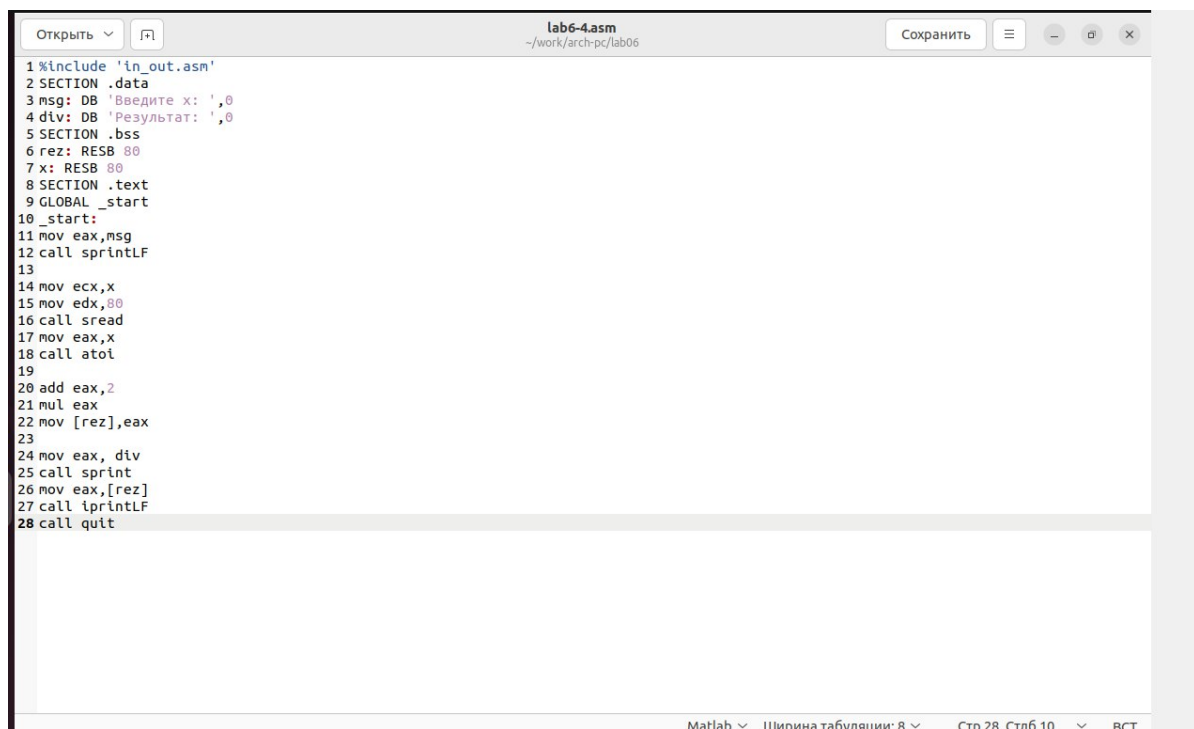
1. Строка “mov eax,rem” и строка “call sprint” отвечают за вывод на экран сообщения ‘Ваш вариант:’.
2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре esx, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки.
3. Инструкция “call atoi” используется для преобразования строки в целое число. Она принимает адрес строки в регистре eax и возвращает полученное число в регистре eax.
4. Строка “xor edx,edx” обнуляет регистр edx перед выполнением деления. Строка “mov ebx,20” загружает значение 20 в регистр ebx. Строка “div ebx”

выполняет деление регистра `eax` на значение регистра `ebx` с сохранением частного в регистре `eax` и остатка в регистре `edx`.

5. Остаток от деления записывается в регистр `edx`.
6. Инструкция `inc edx` используется для увеличения значения в регистре `edx` на 1. В данном случае, она увеличивает остаток от деления на 1.
7. Строка `mov eax,edx` передает значение остатка от деления в регистр `eax`. Строка `call iprintLF` вызывает процедуру `iprintLF` для вывода значения на экран вместе с переводом строки.

### 3.4 Задание для самостоятельной работы

Создаем новый файл в каталоге

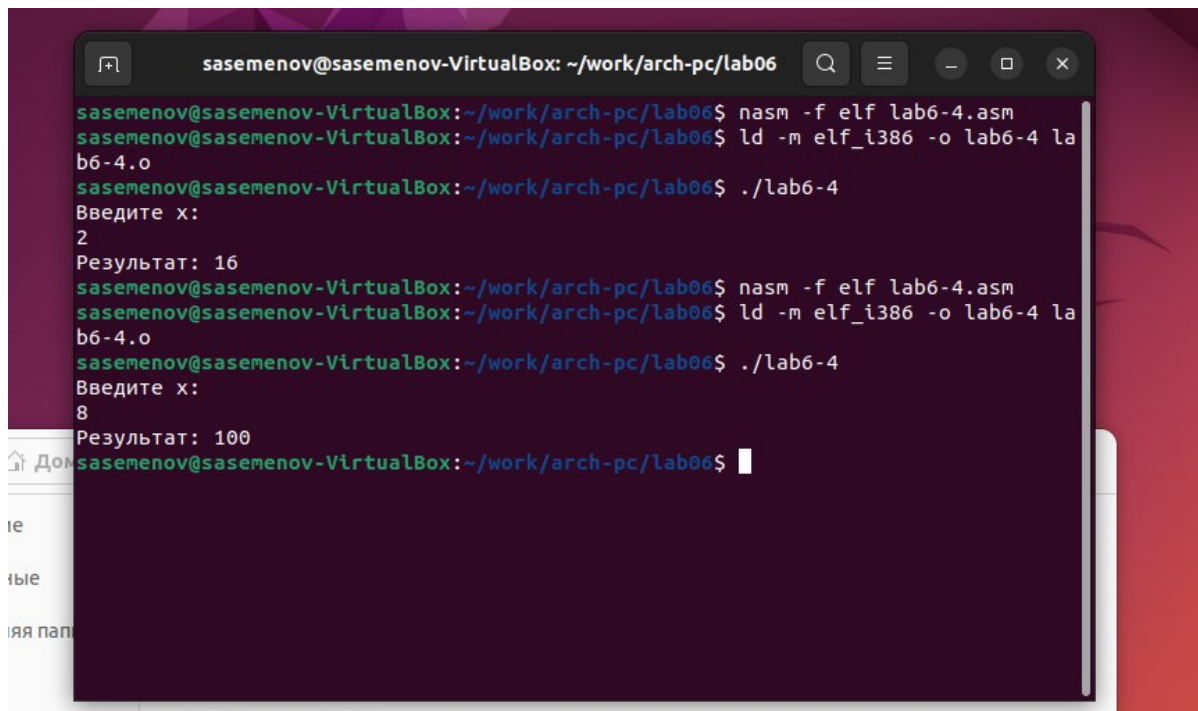


```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите x: ',0
4 div: DB 'Результат: ',0
5 SECTION .bss
6 rez: RESB 80
7 x: RESB 80
8 SECTION .text
9 GLOBAL _start
10 _start:
11 mov eax,msg
12 call sprintf
13
14 mov ecx,x
15 mov edx,80
16 call sread
17 mov eax,x
18 call atoi
19
20 add eax,2
21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

Рис. 3.15: Создаем файл

Открываем его и заполняем, чтобы решалось выражение  $(2+x)^2$

Компилируем программу и проверяем для  $x=2$  и для  $x=8$



```
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab06
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 la
b6-4.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
2
Результат: 16
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 la
b6-4.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
8
Результат: 100
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 3.16: Проверяем работу программы

## 4 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.