

Отчёта по лабораторной работе №6

Освоение арифметических инструкций языка ассемблера NASM.

Кристина Михайловна Салькова

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	20

Список иллюстраций

3.1	Пример программы	7
3.2	Работа программы	8
3.3	Пример программы	8
3.4	Работа программы	9
3.5	Пример программы	10
3.6	Работа программы	10
3.7	Пример программы	11
3.8	Работа программы	11
3.9	Работа программы	12
3.10	Пример программы	13
3.11	Работа программы	13
3.12	Пример программы	14
3.13	Работа программы	15
3.14	Пример программы	16
3.15	Работа программы	16
3.16	Пример программы	18
3.17	Работа программы	19

Список таблиц

1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

2 Задание

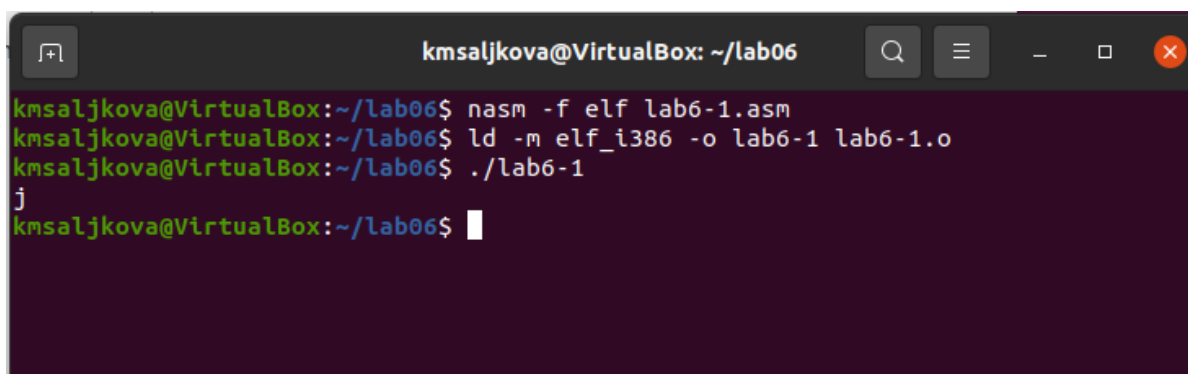
1. Написать программу вычисления выражения $x = x(x)$. Вид функции $x(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы.
2. Загрузите файлы на GitHub.

3 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab7-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax. (рис. 3.1, 3.2)

A screenshot of a text editor window with a dark theme. The title bar shows 'lab6...' and the file path '~/.lab06'. The editor contains assembly code with line numbers 1 through 13. The code includes a directive to include 'in_out.asm', defines a .bss section with a buffer 'buf1' of 80 bytes, defines a .text section, and marks the start of the program. The main logic moves the character '6' into the EAX register, moves the character '4' into the EBX register, adds EBX to EAX, and then prints the contents of EAX using 'sprintf' and 'quit'.

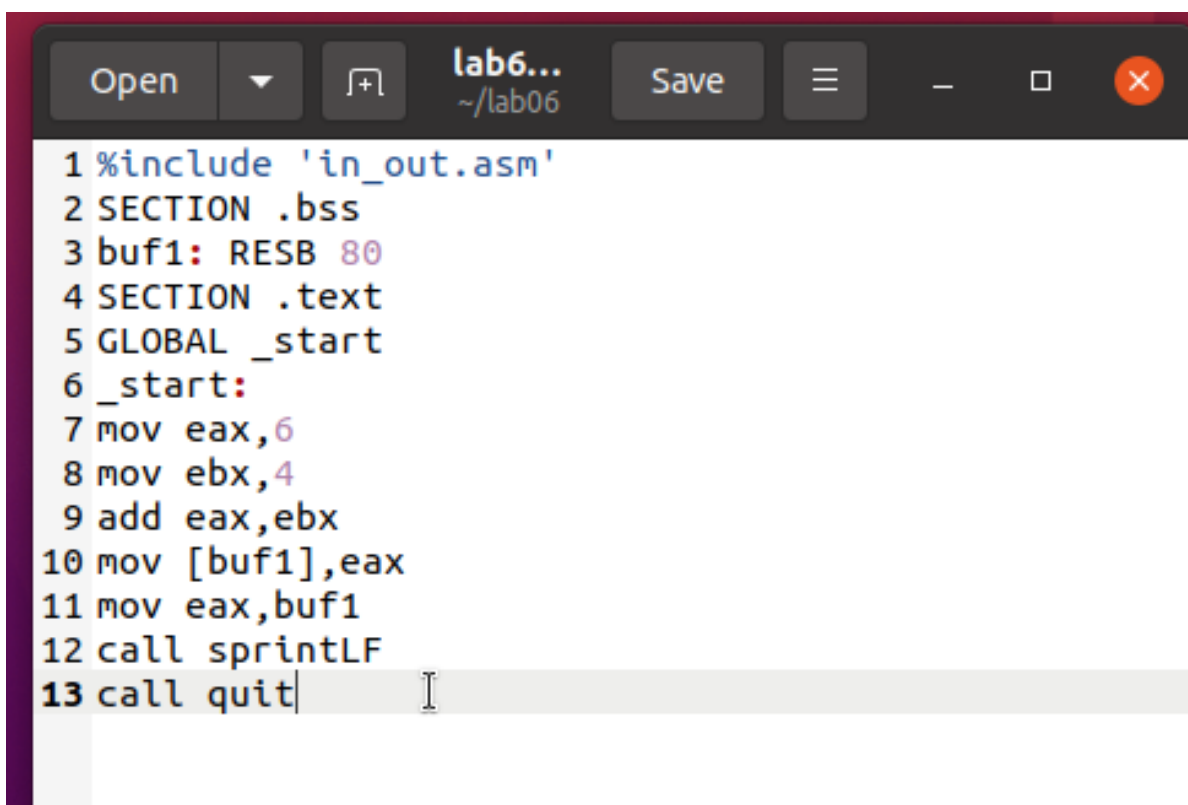
Рис. 3.1: Пример программы

A terminal window titled 'kmsaljkova@VirtualBox: ~/lab06'. It contains the following commands and their outputs:

```
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-1.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-1
j
kmsaljkova@VirtualBox:~/lab06$
```

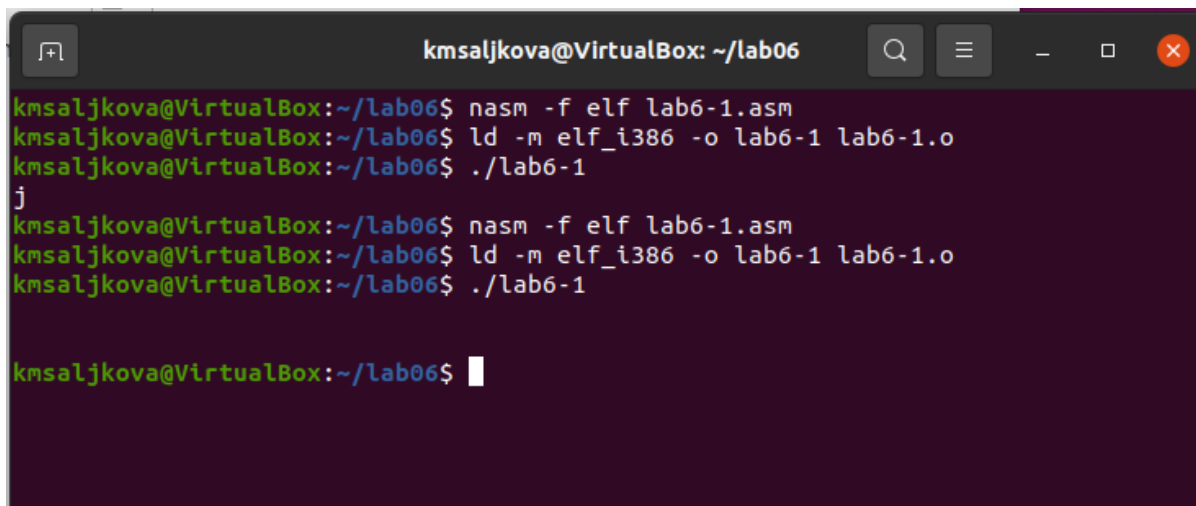
Рис. 3.2: Работа программы

3. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 1) следующим образом: (рис. 3.3, 3.4)

A text editor window titled 'lab6...' with a toolbar containing 'Open', 'Save', and other icons. The code is as follows:

```
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 sprintLF
13 call quit
```

Рис. 3.3: Пример программы

A terminal window titled 'kmsaljkova@VirtualBox: ~/lab06' with standard window controls. The terminal shows a sequence of commands: 'nasm -f elf lab6-1.asm', 'ld -m elf_i386 -o lab6-1 lab6-1.o', and './lab6-1'. The output of the last command is a single character 'j'. The commands are repeated once more.

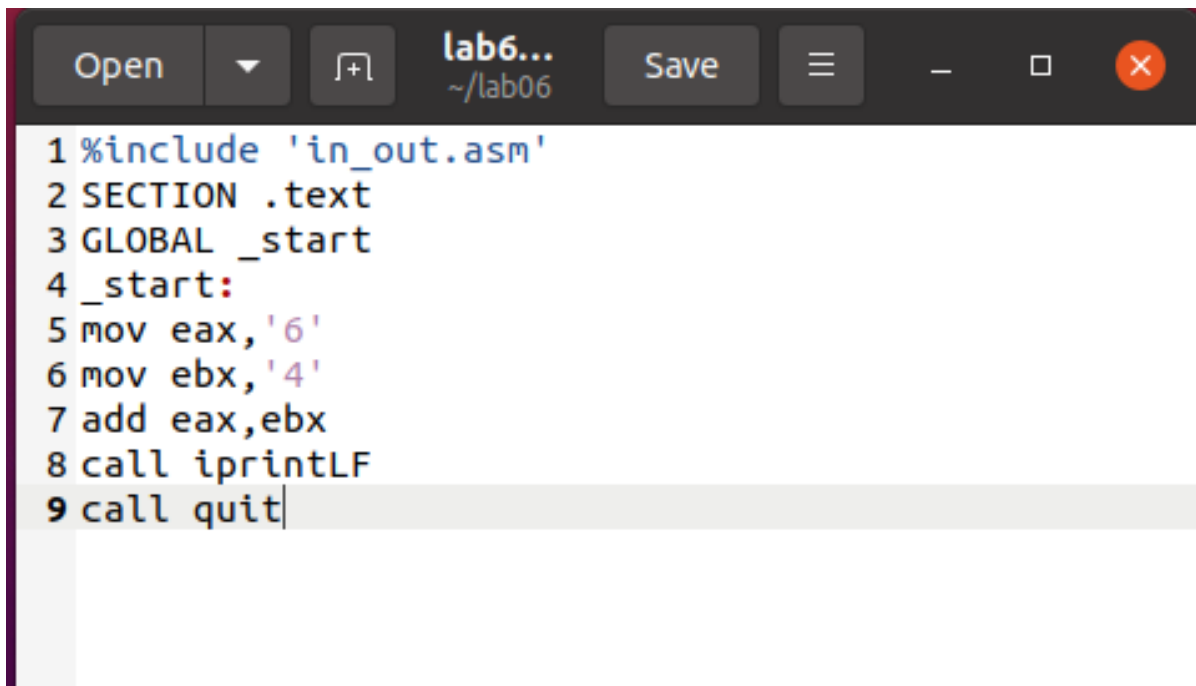
```
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-1.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-1
j
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-1.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-1

kmsaljkova@VirtualBox:~/lab06$
```

Рис. 3.4: Работа программы

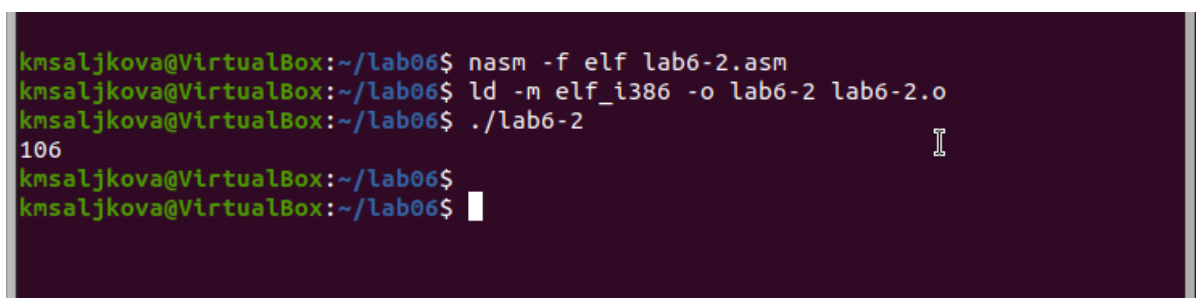
Никакой символ не виден, но он есть. Это возврат каретки LF.

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. 3.5, 3.6)



```
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. 3.5: Пример программы



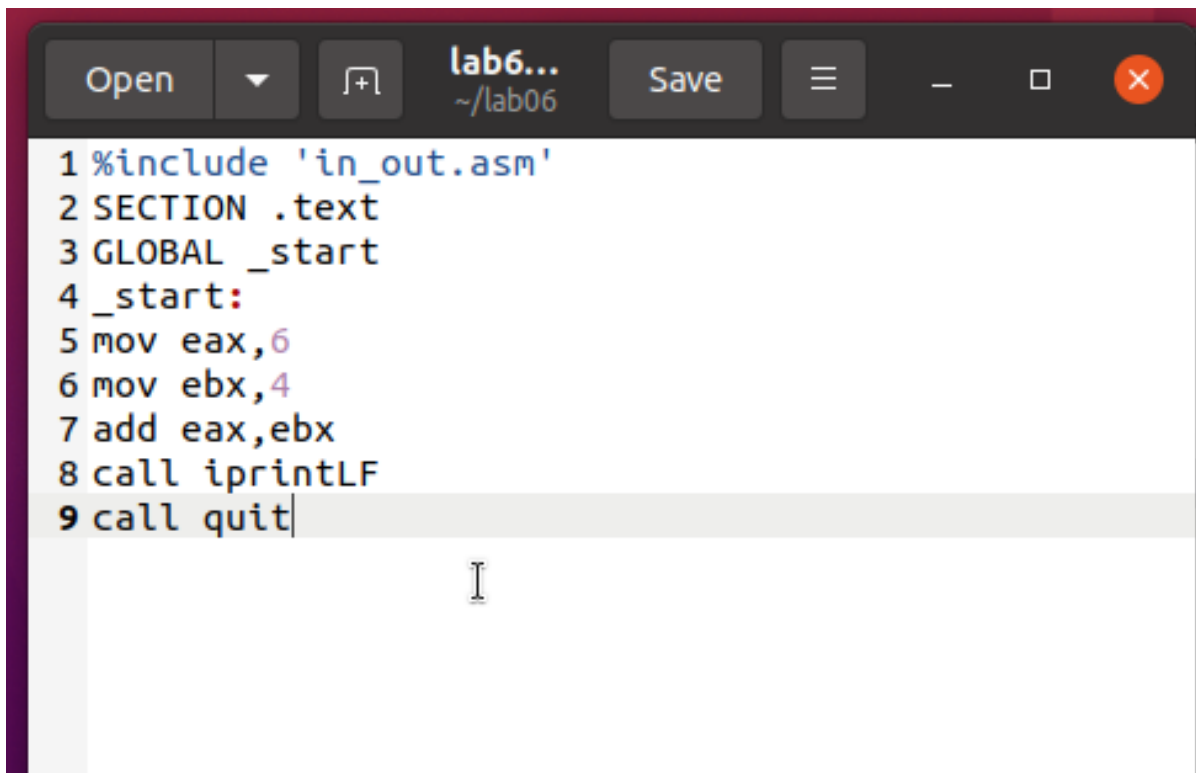
```
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-2
106
kmsaljkova@VirtualBox:~/lab06$
```

Рис. 3.6: 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. 3.7, 3.8)

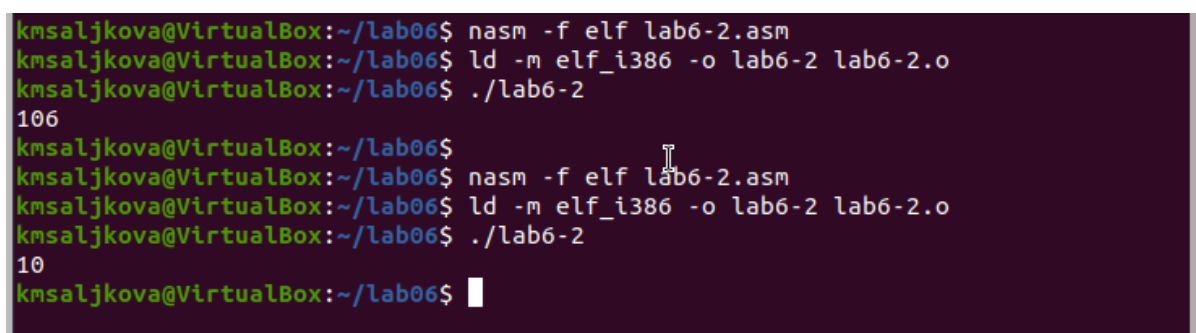
Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы? – получили число 10

A screenshot of a text editor window titled 'lab6...' with a dark theme. The window contains assembly code for a program. The code is as follows:

```
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
```

The cursor is positioned at the end of line 9.

Рис. 3.7: Пример программы

A screenshot of a terminal window showing the compilation and execution of the assembly program. The commands and their outputs are as follows:

```
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-2
106
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-2
10
kmsaljkova@VirtualBox:~/lab06$
```

Рис. 3.8: 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. 3.9)

```

kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-2
106
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-2
10
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-2.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-2
10kmsaljkova@VirtualBox:~/lab06$

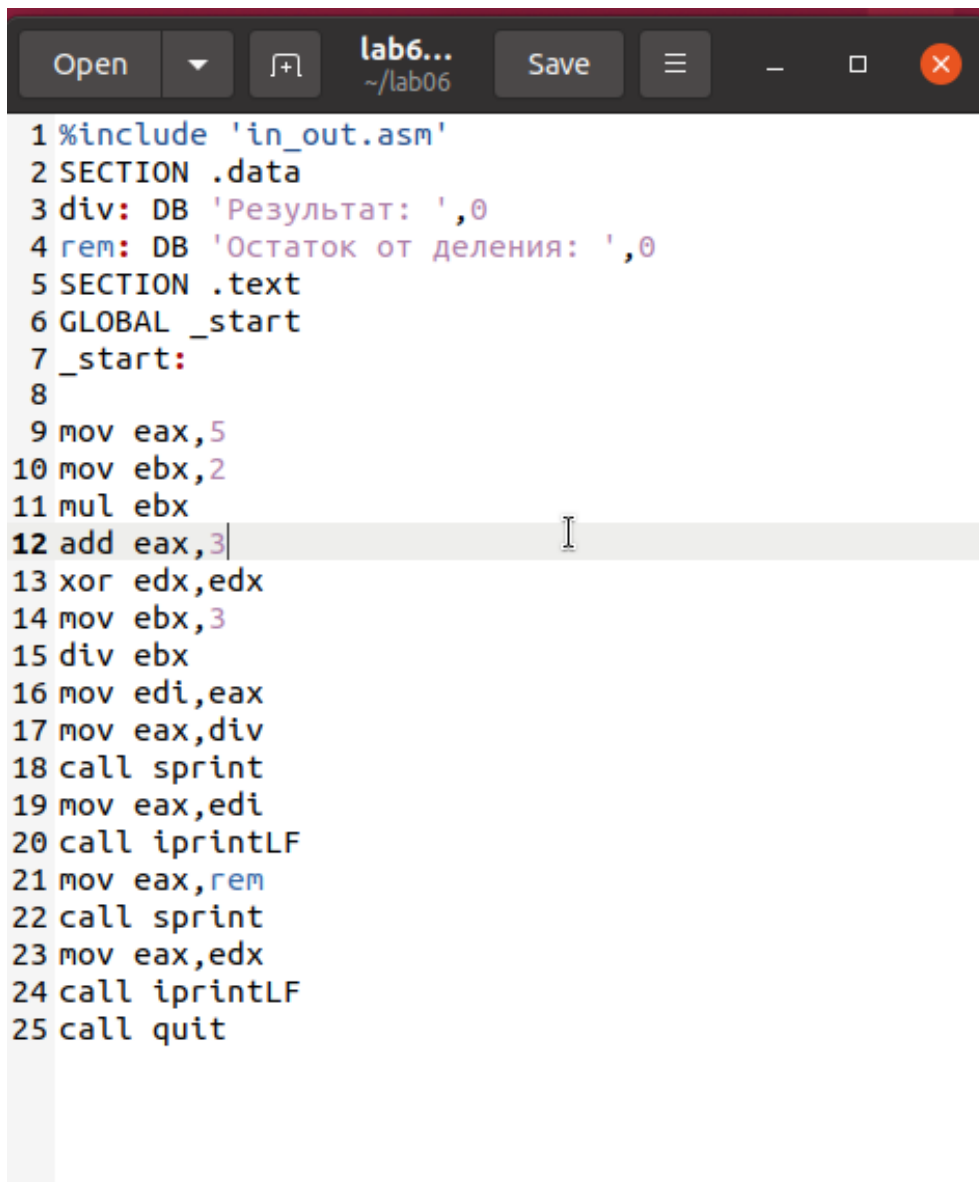
```

Рис. 3.9: Работа программы

6. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения

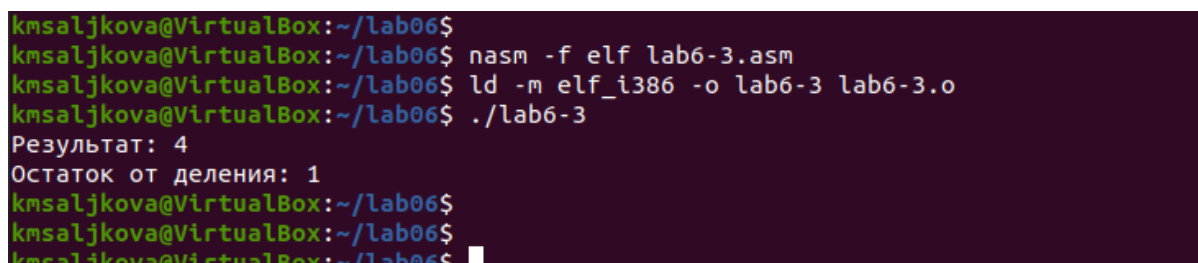
$$f(x) = (5 * 2 + 3) / 3$$

. (рис. 3.10, рис. 3.11)



```
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. 3.10: Пример программы



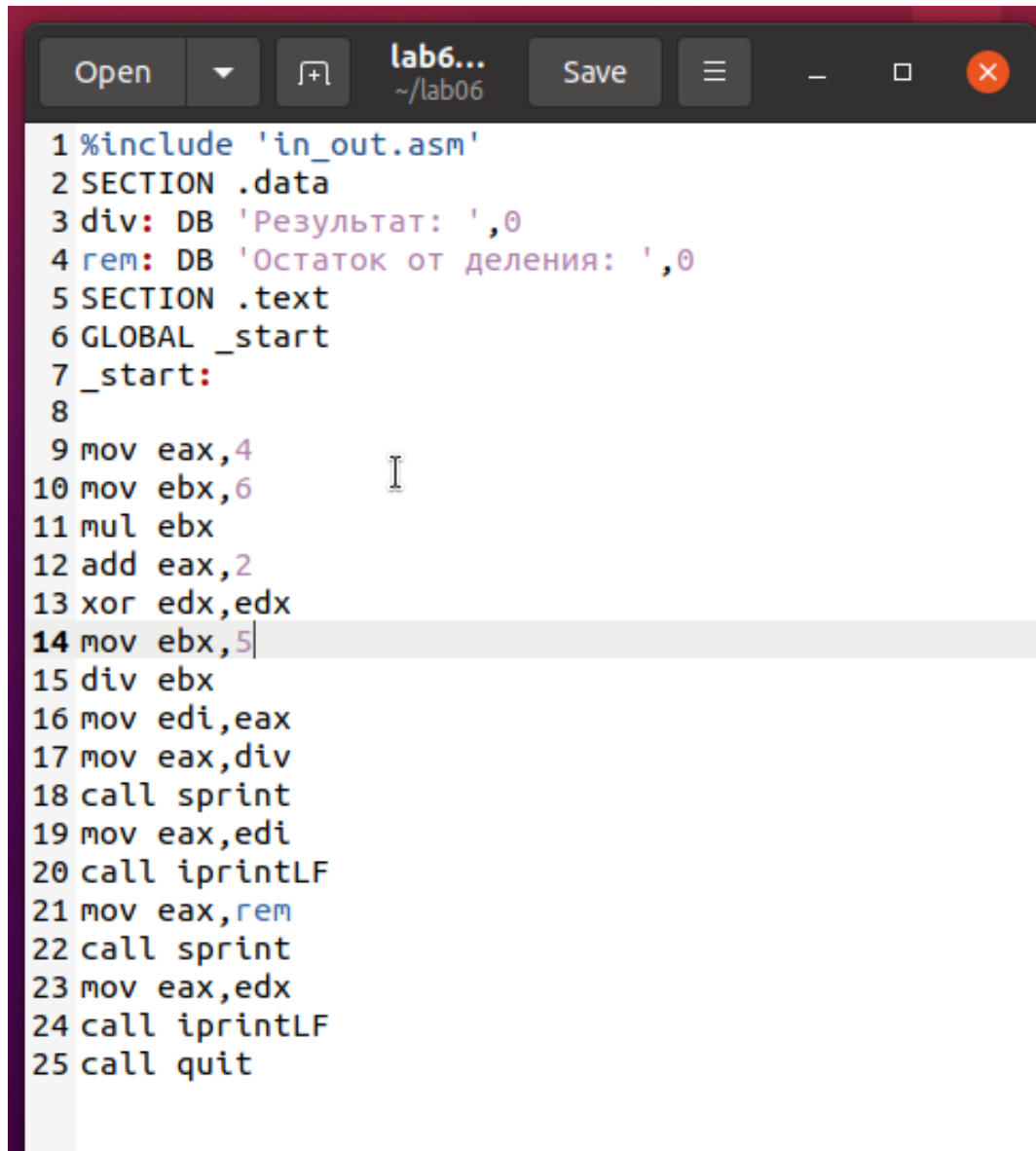
```
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-3.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$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

Рис. 3.11: Работа программы

Измените текст программы для вычисления выражения

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

. Создайте исполняемый файл и проверьте его работу. (рис. 3.12, рис. 3.13)



```
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. 3.12: Пример программы

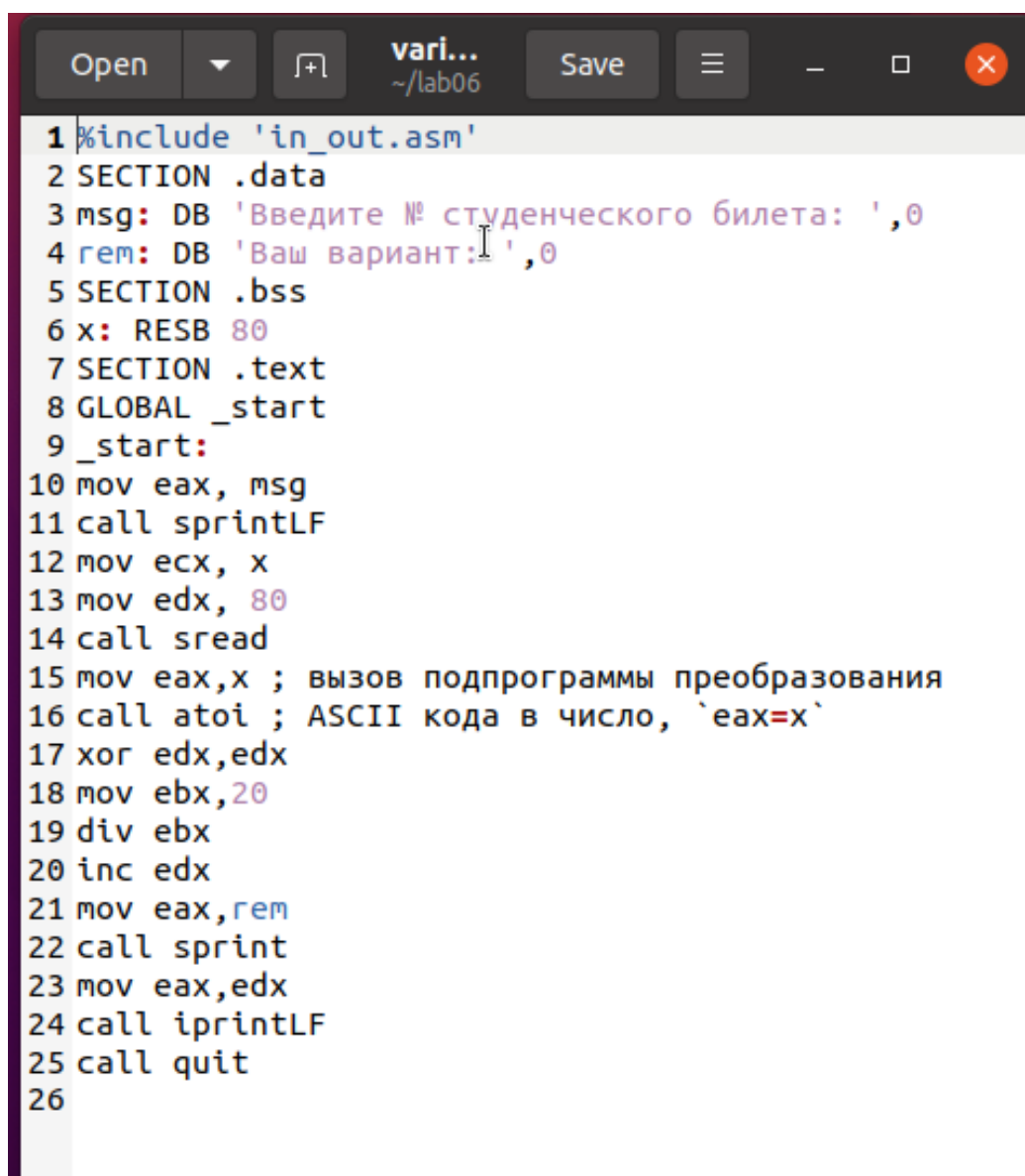
```

kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-3.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf lab6-3.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
kmsaljkova@VirtualBox:~/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$

```

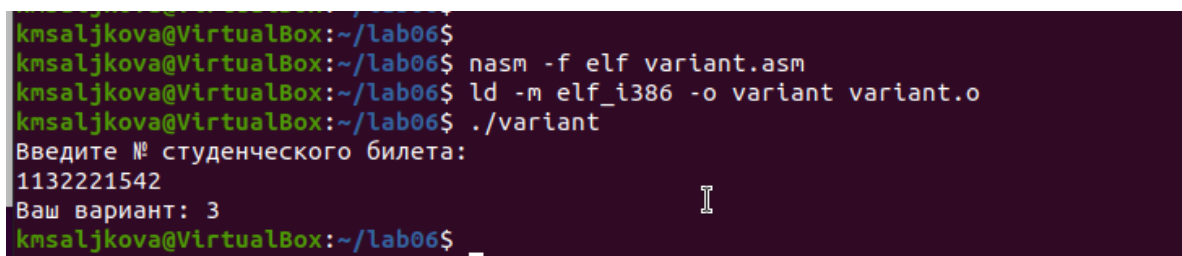
Рис. 3.13: Работа программы

7. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму: (рис. 3.14, рис. 3.15)



```
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 sprintf
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x ; вызов подпрограммы преобразования
16 call atoi ; ASCII кода в число, `eax=x`
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
26
```

Рис. 3.14: Пример программы



```
kmsaljkova@VirtualBox:~/lab06$
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf variant.asm
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o variant variant.o
kmsaljkova@VirtualBox:~/lab06$ ./variant
Введите № студенческого билета:
1132221542
Ваш вариант: 3
kmsaljkova@VirtualBox:~/lab06$
```

Рис. 3.15: 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? `push ecx, x` `mov edx, 80` `call sread`

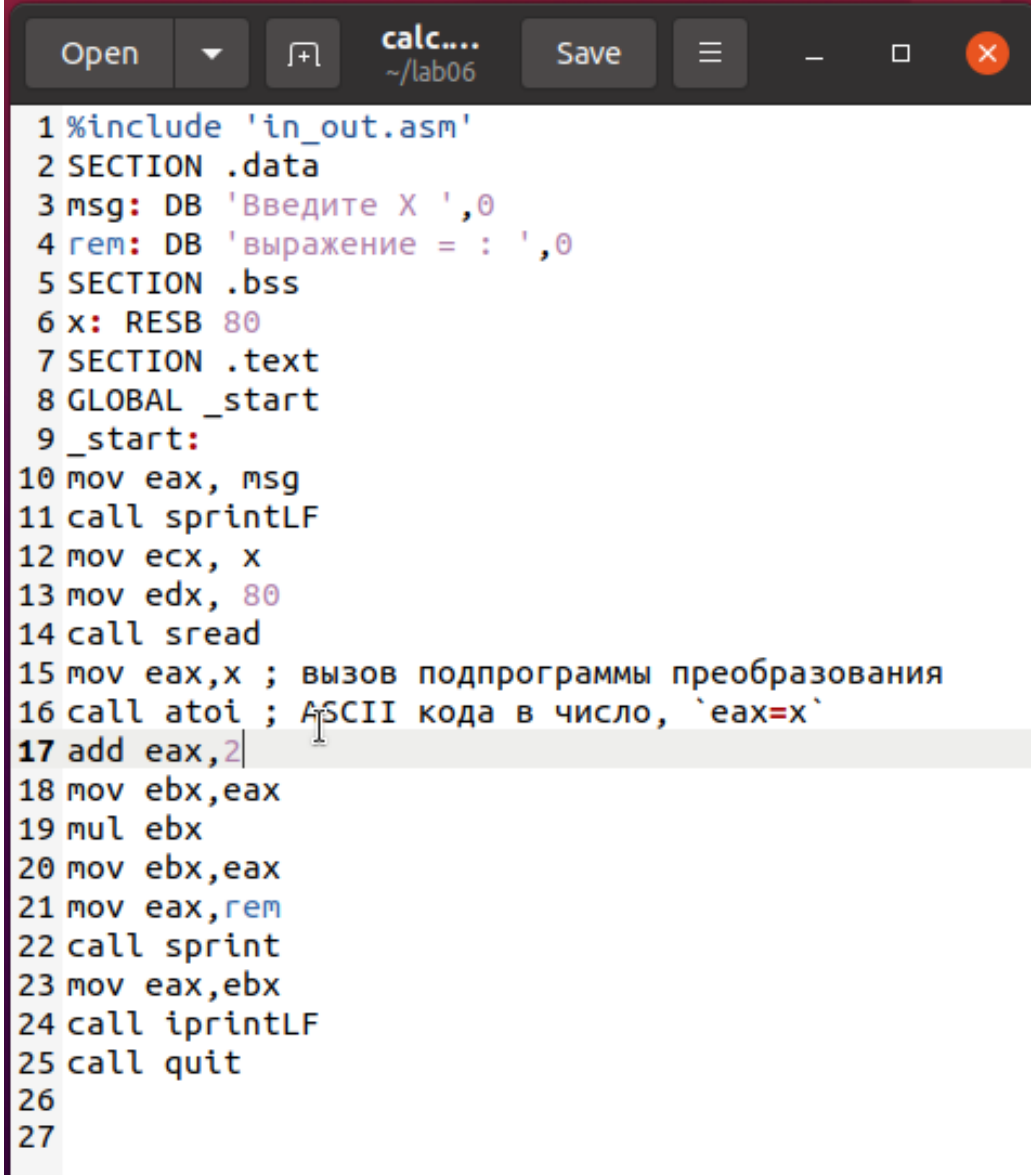
Считывает значение студбилета в переменную X из консоли

- 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 вычисления? `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. 3.16, рис. 3.17)

Получили вариант 3 -

$$(x + 2)^2$$

для $x=2$ и 8

A screenshot of a text editor window titled 'calc....' with a path '~/.lab06'. The window contains assembly code for a program that calculates (x+2)^2. The code includes sections for data, bss, and text, and uses various assembly instructions like mov, call, add, mul, and iprintLF. The code is numbered from 1 to 27.

```
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 ; вызов подпрограммы преобразования
16 call atoi ; ASCII кода в число, `eax=x`
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
27
```

Рис. 3.16: Пример программы

```
kmsaljkova@VirtualBox:~/lab06$  
kmsaljkova@VirtualBox:~/lab06$  
kmsaljkova@VirtualBox:~/lab06$ nasm -f elf calc.asm  
kmsaljkova@VirtualBox:~/lab06$ ld -m elf_i386 -o calc calc.o  
kmsaljkova@VirtualBox:~/lab06$ ./calc  
Введите X  
2  
выражение = : 16  
kmsaljkova@VirtualBox:~/lab06$ ./calc  
Введите X  
8  
выражение = : 100  
kmsaljkova@VirtualBox:~/lab06$  
kmsaljkova@VirtualBox:~/lab06$
```

Рис. 3.17: Работа программы

4 Выводы

В ходе выполнения работы, я освоила работу с арифметическими операциями