

# **Отчёт по лабораторной работе №6**

**Арифметические операции в NASM**

Малкина Дарья Александровна

# 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 . . . . .	11
2.3	Ответы на вопросы . . . . .	13
<b>3</b>	<b>Выполнение задания для самостоятельной работы</b>	<b>15</b>
<b>4</b>	<b>Выводы</b>	<b>17</b>

## Список иллюстраций

2.1	Вводим текст программы lab6-1 . . . . .	7
2.2	Результат программы - j . . . . .	7
2.3	Редактируем текст программы lab6-1 . . . . .	8
2.4	Результат программы - перевод строки . . . . .	9
2.5	Программа lab6-2 с функциями внешнего файла . . . . .	9
2.6	Результат программы - 106 . . . . .	10
2.7	Редактируем текст программы lab6-2 . . . . .	10
2.8	Результат программы - 10 . . . . .	11
2.9	Результат программы . . . . .	11
2.10	Вводим текст программы lab6-3 . . . . .	12
2.11	Результат вычисления $f(x) = (5 \cdot 2 + 3)/3$ . . . . .	12
2.12	Редактируем текст программы lab6-3 . . . . .	13
2.13	Результат вычисления $f(x) = (4 \cdot 6 + 2)/5$ . . . . .	13
2.14	Результат программы var . . . . .	13
3.1	Вводим текст сообщений . . . . .	15
3.2	Старт программы, запись и преобразование строки с клавиатуры .	15
3.3	Вычисление выражения . . . . .	16
3.4	Вывод результата и завершение программы . . . . .	16
3.5	Результат программы var5 . . . . .	16

## **Список таблиц**

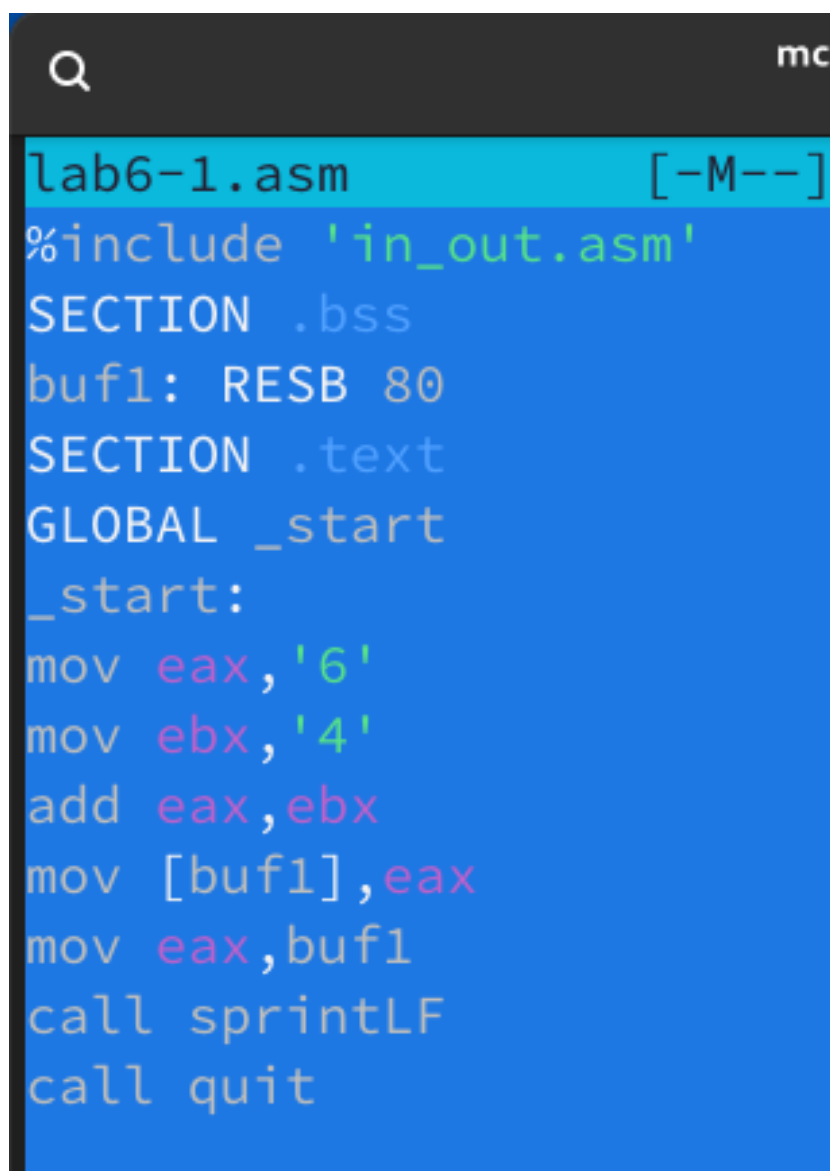
# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## **2 Выполнение лабораторной работы**

### **2.1 Символьные и численные данные в NASM**

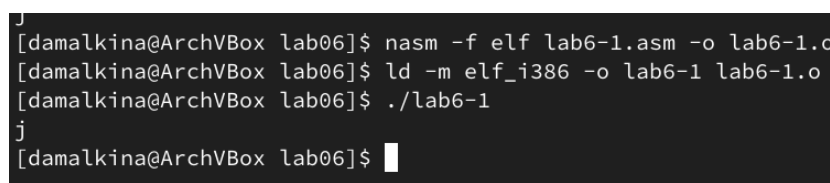
1. Создадим каталог для программ лабораторной работы № 6, в нем создадим файл `lab6-1.asm`, в который введём текст программы из листинга 6.1.:



```
lab6-1.asm [-M--]
#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: Вводим текст программы lab6-1

Создаём исполняемый файл и запускаем его, программа выводит символ j:



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

Рис. 2.2: Результат программы - j

3. Далее изменим текст программы и вместо символов, запишем в регистры

числа:

```
1 %include 'in_out.asm'
2
3 SECTION .bss
4 buf1: RESB 80
5
6 SECTION .text
7 GLOBAL _start
8 _start:
9
10 mov eax, 6
11 mov ebx, 4
12 add eax, ebx
13 mov [buf1], eax
14 mov eax, buf1
15 call sprintf
16
17 call quit
```

Рис. 2.3: Редактируем текст программы lab6-1

Создаём исполняемый файл и запускаем его, программа выводит символ с кодом 10. Пользуясь таблицей ASCII определяем, что коду 10 соответствует символу перевода строки, что мы и наблюдаем в результате:

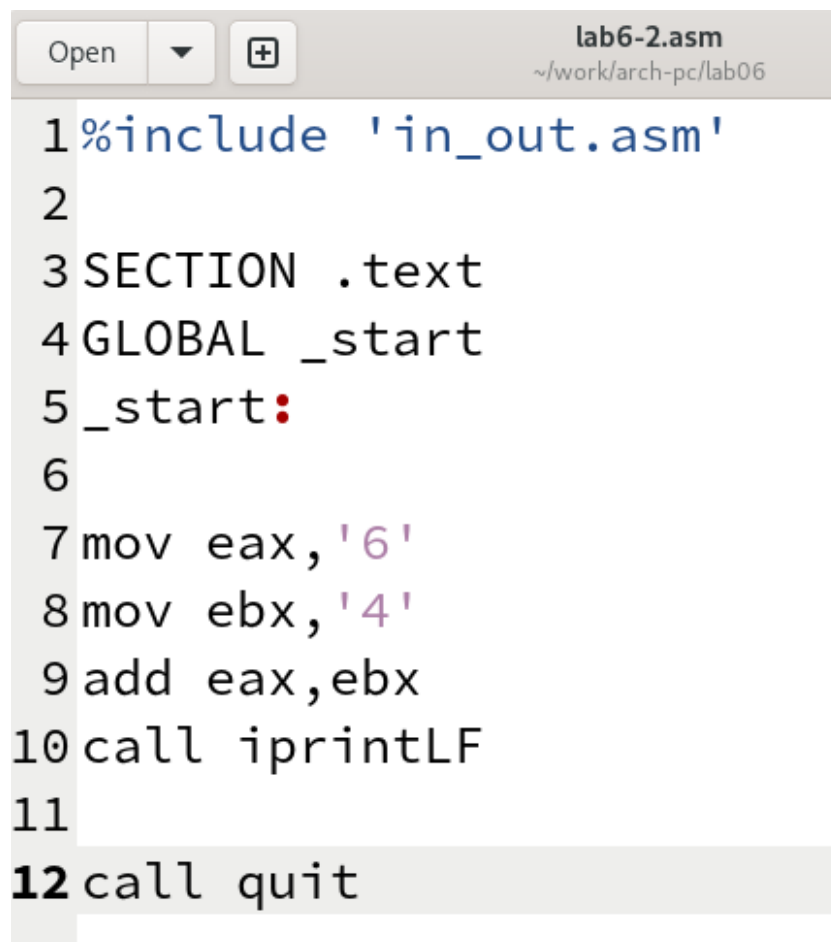


```
[damalkina@ArchVBox lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[damalkina@ArchVBox lab06]$ nasm -f elf lab6-1.asm -o lab6-1.o
[damalkina@ArchVBox lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[damalkina@ArchVBox lab06]$ ./lab6-1

[damalkina@ArchVBox lab06]$
```

Рис. 2.4: Результат программы - перевод строки

4. Создаём файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него текст программы, преобразованный с использованием функций из внешнего файла in\_out.asm:



```
lab6-2.asm
~/work/arch-pc/lab06

1 %include 'in_out.asm'
2
3 SECTION .text
4 GLOBAL _start
5 _start:
6
7 mov eax, '6'
8 mov ebx, '4'
9 add eax, ebx
10 call iprintLF
11
12 call quit
```

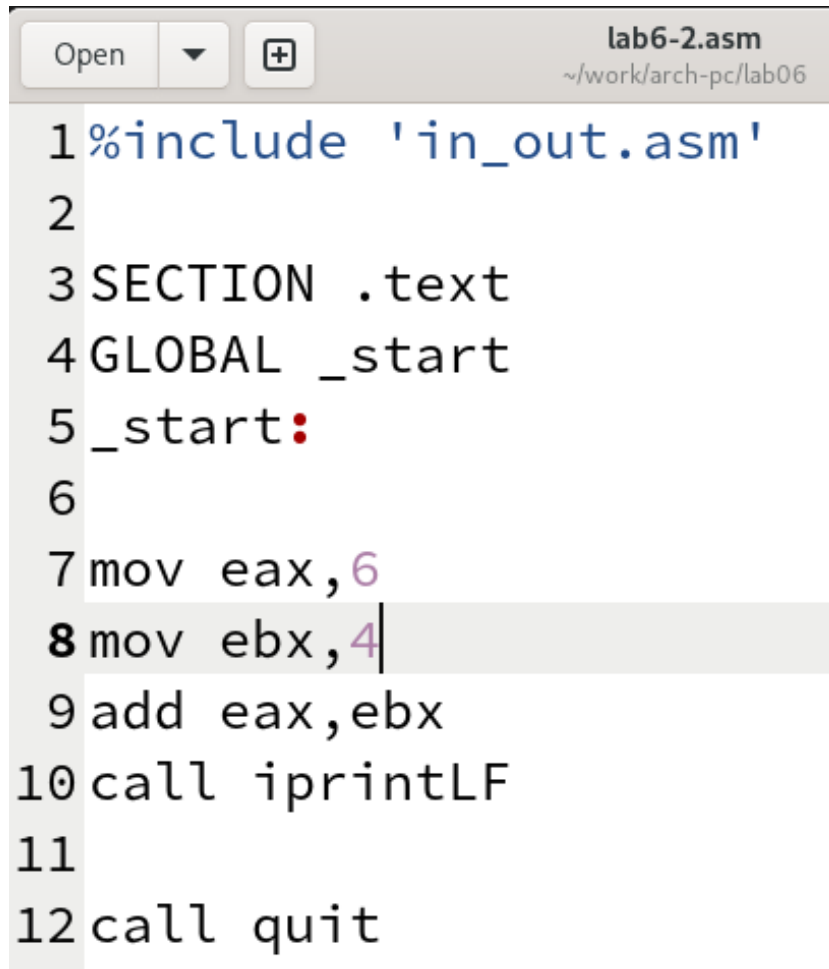
Рис. 2.5: Программа lab6-2 с функциями внешнего файла

В результате работы программы получаем число 106:

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

Рис. 2.6: Результат программы - 106

5. Изменим текст программы - вместо символов, запишем в регистры числа:



```
lab6-2.asm
~/work/arch-pc/lab06

1 %include 'in_out.asm'
2
3 SECTION .text
4 GLOBAL _start
5 _start:
6
7 mov eax, 6
8 mov ebx, 4
9 add eax, ebx
10 call iprintLF
11
12 call quit
```

Рис. 2.7: Редактируем текст программы lab6-2

Создаём исполняемый файл и запускаем его, при исполнении программы получили результат 10:

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

Рис. 2.8: Результат программы - 10

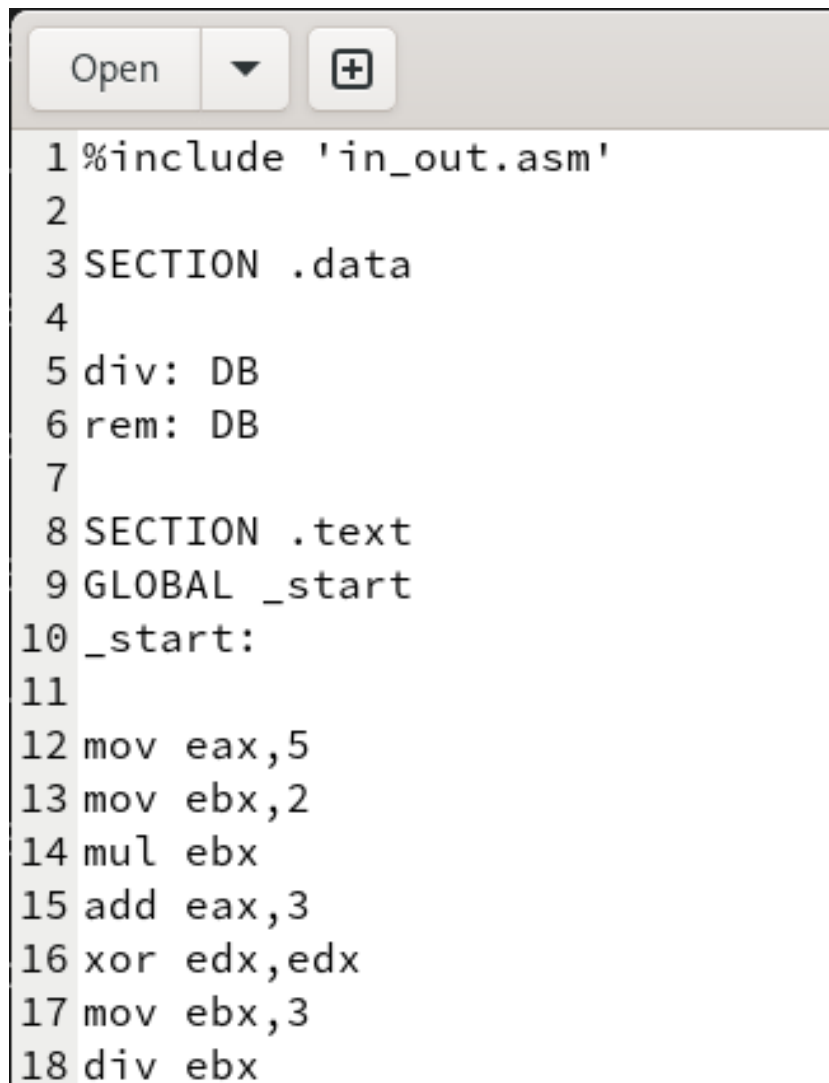
Заменим функцию `iprintLF` на `iprint`, тогда после запуска программа не переводит строку по завершении работы:

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

Рис. 2.9: Результат программы

## 2.2 Выполнение арифметических операций в NASM

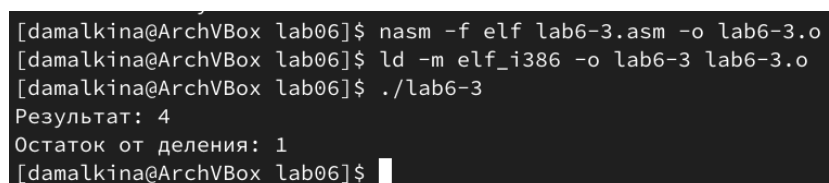
6. Создадим файл `lab6-3.asm` и введем в него программу вычисления арифметического выражения  $f(x) = (5 \cdot 2 + 3)/3$ :



```
1 %include 'in_out.asm'
2
3 SECTION .data
4
5 div: DB
6 rem: DB
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 mov eax,5
13 mov ebx,2
14 mul ebx
15 add eax,3
16 xor edx,edx
17 mov ebx,3
18 div ebx
```

Рис. 2.10: Вводим текст программы lab6-3

Создаём исполняемый файл и запускаем его:



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

Рис. 2.11: Результат вычисления  $f(x) = (5 \cdot 2 + 3)/3$

Теперь изменим текст программы, чтобы получить результат вычисления выражения  $f(x) = (4 \cdot 6 + 2)/5$ :

```

18 mov eax,4 ; EAX=4
19 mov ebx,6 ; EBX=6
20 mul ebx ; EAX=EAX*EBX
21 add eax,2 ; EAX=EAX+2
22 xor edx,edx ; обнуляем EDX для корректной работы div
23 mov ebx,5 ; EBX=5
24 div ebx ; EAX=EAX/3, EDX=остаток от деления

```

Рис. 2.12: Редактируем текст программы lab6-3

Создаём исполняемый файл и запускаем его:

```

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

```

Рис. 2.13: Результат вычисления  $f(x) = (4 \cdot 6 + 2)/5$

7. Создадим файл variant.asm, введем в файл текст программы, которая будет вычислять вариант задания по номеру студенческого билета. Создаем исполняемый файл var и запускаем его:

```

[damalkina@ArchVBox lab06]$ nasm -f elf variant.asm -o var.o
[damalkina@ArchVBox lab06]$ ld -m elf_i386 -o var var.o
[damalkina@ArchVBox lab06]$ ./var
Введите № студенческого билета:
1132246784
Ваш вариант: 5
[damalkina@ArchVBox lab06]$

```

Рис. 2.14: Результат программы var

Проверим результат аналитически:  $56612339 \cdot 20 = 1132246780$   $1132246784 - 1132246780 = 4$  (остаток от деления) Получаем  $4 + 1 = 5$  вариант задания.

## 2.3 Ответы на вопросы

1. За вывод на экран сообщения отвечают следующие строки:

rem: DB 'Ваш вариант:',0 ... mov eax,rem call sprint

2. Строки кода ниже отвечают за считывание строки с клавиатуры и сохранение ее в буфер для дальнейшей обработки:

mov ecx, x mov edx, 80 call sread

3. Инструкция “call atoi” используется для вызова функции “atoi”, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax.

4. За вычисления варианта отвечают следующие строки кода:

mov eax,x call atoi ... xor edx,edx mov ebx,20 div ebx inc edx

5. Остаток от деления при выполнении инструкции “div ebx” записывается в регистр edx.

6. Инструкция “inc edx” увеличивает значение в регистре edx на 1. То есть, после деления значение в edx будет остаток от деления, к этому остатку добавляется 1, чтобы получить вариант.

7. За вывод на экран результата вычислений отвечают строки:

mov eax,rem call sprint mov eax,edx call iprintLF

### 3 Выполнение задания для самостоятельной работы

Напишем программу вычисления выражения  $y = f(x) = (9 \cdot x - 8)/8$ . Возьмём за основу текст программы var. Создадим копию файла var.asm файл var5.asm. Внесём изменения:

```
1 ;-----
2 ; Программа вычисления выражения
3 ;-----
4
5 %include 'in_out.asm'
6
7 SECTION .data
8 msg: DB 'Вариант №5 (9x - 8)/8. Введите значение переменной x: ',0
9 rem: DB 'Результат: ',0 ; Сообщение для вывода результата
10
11
12 SECTION .bss
13 x: RESB 80 ; Буфер для хранения введенной строки
14
15 SECTION .text
16 GLOBAL _start
```

Рис. 3.1: Вводим текст сообщений

```
18 _start:
19
20 mov eax, msg ; Загружаем адрес сообщения для ввода x в регистр eax
21 call sprint ; Выводим сообщение на экран
22
23 mov ecx, x ; Загружаем адрес буфера x в регистр ecx
24 mov edx, 80 ; Устанавливаем максимальную длину вводимой строки
25 call sread ; Считываем строку с клавиатуры и сохраняем ее в буфер x
26
27 mov eax, x ; Загружаем адрес буфера x в регистр eax
28 call atoi ; Преобразуем ASCII-код введенной строки в число, результат в eax
```

Рис. 3.2: Старт программы, запись и преобразование строки с клавиатуры

```

30 ;-----
31 ; Вычисление выражения (9*x - 8)/8
32 ;-----
33
34 xor edx,edx      ; Обнуляем регистр edx перед делением
35 mov ebx,9        ; Загружаем 9 в регистр ebx
36 mul ebx          ; Умножаем значение в eax (x) на 9
37 sub eax,8        ; Вычитаем 8 из eax
38 mov ebx,8        ; Загружаем 8 в регистр ebx
39 div ebx          ; Делим на 8
40 mov edx,eax      ; Сохраняем результат в edx

```

Рис. 3.3: Вычисление выражения

```

42 ;-----
43 ; Вывод результата вычисления
44 ;-----
45
46 mov eax,rem      ; Загружаем адрес сообщения 'Результат: '
47 call sprint      ; Выводим сообщение 'Результат: '
48 mov eax,edx      ; Загружаем результат вычисления в eax
49 call iprintLF    ; Выводим результат на экран с переводом строки
50
51 call quit        ; Завершаем программу

```

Рис. 3.4: Вывод результата и завершение программы

Создаём исполняемый файл и проверяем его работу для значений  $x_1=8$  и  $x_2=64$ :

```

[damalkina@ArchVBox lab06]$ nasm -f elf var5.asm -o var5.o
[damalkina@ArchVBox lab06]$ ld -m elf_i386 -o var5 var5.o
[damalkina@ArchVBox lab06]$ ./var5
Вариант №5 (9x - 8)/8. Введите значение переменной x: 64
Результат: 71
[damalkina@ArchVBox lab06]$ ./var5
Вариант №5 (9x - 8)/8. Введите значение переменной x: 8
Результат: 8
[damalkina@ArchVBox lab06]$ 

```

Рис. 3.5: Результат программы var5

Проверим работу программы посчитав аналитически значения выражения при  $x_1=8$  и  $x_2=64$ :

$$x_1=8: (9 \cdot 8 - 8)/8 = 64/8 = 8 \quad x_2=64: (9 \cdot 64 - 8)/8 = 8 \cdot (72 - 1)/8 = 71$$

Программа работает верно.



## 4 Выводы

В ходе выполнения лабораторной работы мы узнали об арифметических инструкциях языка ассемблера NASM и научились их использовать при написании текста программы. Также научились создавать ассемблерную программу, которая считывает данные от пользователя, выполняет арифметические операции и выводит результат на экран. Мы узнали, как выполнять операции умножения и деления, а также как использовать подпрограммы для ввода-вывода. Кроме того, мы научились правильно обрабатывать данные, которые пользователь вводит с клавиатуры, преобразуя их из ASCII-кода в числовое значение.