

# **Лабораторная работа №6**

**Арифметические операции в NASM**

Глушенок Анна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Ход выполнения работы</b>	<b>6</b>
2.1	Задание 1 . . . . .	6
2.2	Задание 2 . . . . .	11
2.3	Ответы на вопросы . . . . .	16
2.4	Задания для самостоятельной работы . . . . .	17
<b>3</b>	<b>Выводы</b>	<b>20</b>

## Список иллюстраций

2.1	Создание файла для работы . . . . .	6
2.2	Программа вывода значения регистра eax . . . . .	7
2.3	Результат работы программы . . . . .	7
2.4	Внесение изменений в программу . . . . .	8
2.5	Результат работы программы . . . . .	8
2.6	Создание файла для работы . . . . .	9
2.7	Программа вывода значения регистра eax . . . . .	9
2.8	Результат работы программы . . . . .	10
2.9	Замена символов на числа . . . . .	10
2.10	Результат программы с <code>iprintLF</code> . . . . .	11
2.11	Результат программы с <code>iprint</code> . . . . .	11
2.12	Создание файла . . . . .	12
2.13	Программа вычисления выражения $f(x)=(5*2+3)/3$ . . . . .	12
2.14	Результат работы программы . . . . .	13
2.15	Программа вычисления выражения $f(x)=(4*6+2)/5$ . . . . .	13
2.16	Результат работы программы . . . . .	14
2.17	Создание файла для работы . . . . .	14
2.18	Программа вычисления варианта по номеру студ билета . . . . .	15
2.19	Результат работы программы . . . . .	15
2.20	Проверка результата программы . . . . .	16
2.21	Создание программы . . . . .	18
2.22	Проверка работы программы . . . . .	19

## **Список таблиц**

# 1 Цель работы

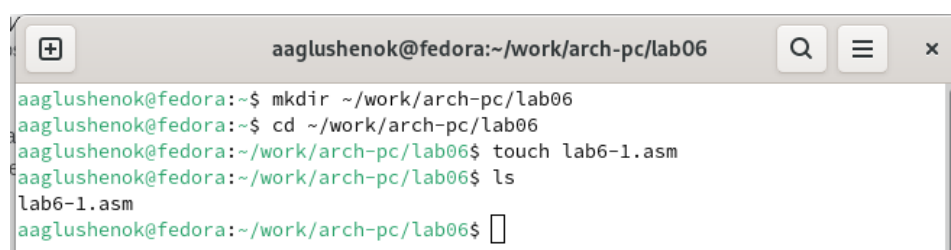
Освоение арифметических инструкций языка ассемблера NASM.

## 2 Ход выполнения работы

### 2.1 Задание 1

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm.

Создаем указанный каталог, используя команду `mkdir`, переходим в него и с помощью команды `touch` создаем нужный файл.

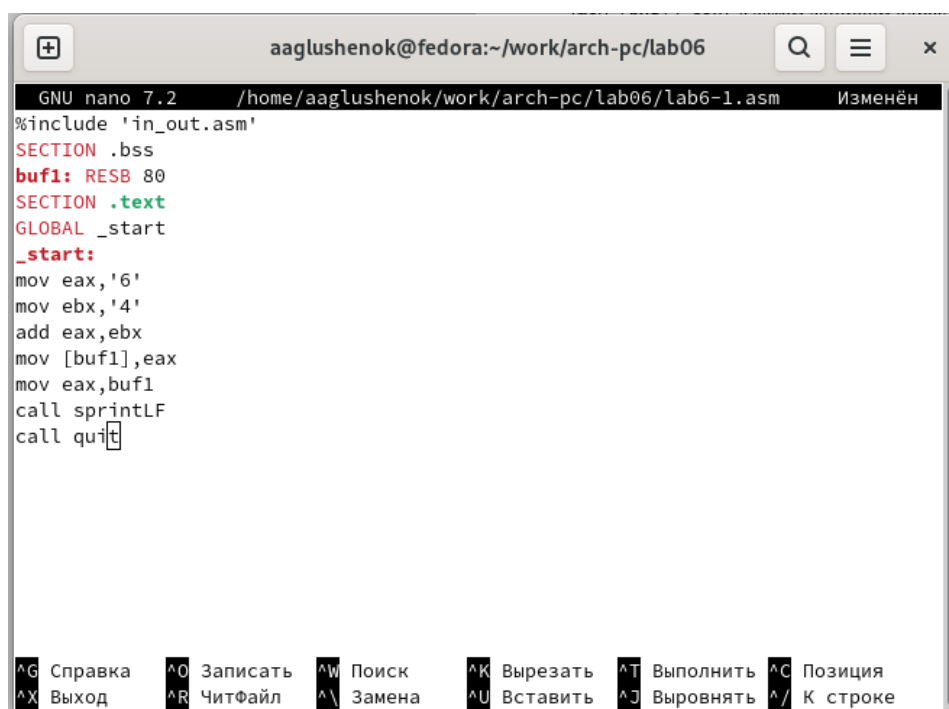


```
aaglushenok@fedora:~/work/arch-pc/lab06$ mkdir ~/work/arch-pc/lab06
aaglushenok@fedora:~/work/arch-pc/lab06$ cd ~/work/arch-pc/lab06
aaglushenok@fedora:~/work/arch-pc/lab06$ touch lab6-1.asm
aaglushenok@fedora:~/work/arch-pc/lab06$ ls
lab6-1.asm
aaglushenok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.1: Создание файла для работы

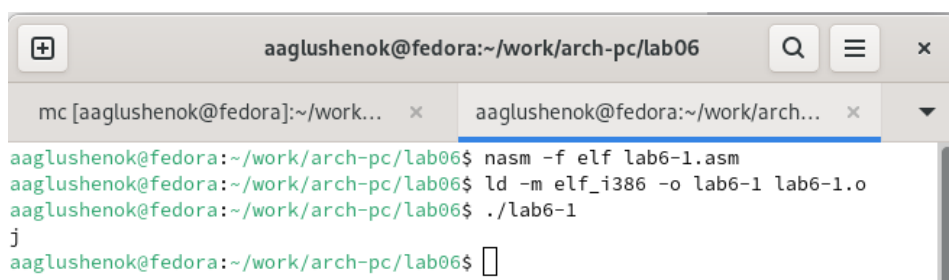
2. Введите в файл lab6-1.asm текст программы из листинга 6.1. (Программа вывода значения регистра `eax`). Создайте исполняемый файл и запустите его.

Вводим текст листинга 6.1., создаем исполняемый файл, и в результате его работы получаем символ “j”.



```
GNU nano 7.2 /home/aaglushmanok/work/arch-pc/lab06/lab6-1.asm
%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. 2.2: Программа вывода значения регистра eax



```
aaglushmanok@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
aaglushmanok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
aaglushmanok@fedora:~/work/arch-pc/lab06$ ./lab6-1
j
aaglushmanok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.3: Результат работы программы

3. Внесите изменения в текст программы: вместо символов запишите в регистры числа. Создайте исполняемый файл и запустите его. Пользуясь таблицей ASCII определите какому символу соответствует код 10.

Вносим изменения в программу: убирая кавычки возле символов, превращаем их в числа. Создаем исполняемый файл, и в результате его работы получаем 2 пустых строки. Так происходит из-за того, что коду 10 в таблице ASCII соответствует символ переноса строки (сама строка остается пустой).

```
mc [aaglushenok@fedora]:~/work/arch-pc/lab06
GNU nano 7.2 /home/aaglushenok/work/arch-pc/lab06/lab6-1.asm
%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 printf
call quit

```

^G Справка   ^O Записать   ^W Поиск   ^K Вырезать   ^T Выполнить   ^C Позиция  
^X Выход   ^R ЧитФайл   ^\ Замена   ^U Вставить   ^J Выводить   ^/\_ К строке

Рис. 2.4: Внесение изменений в программу

```
aaglushenok@fedora:~/work/arch-pc/lab06
mc [aaglushenok@fedora]:~/work/arch-pc/lab06
aaglushenok@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
aaglushenok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
aaglushenok@fedora:~/work/arch-pc/lab06$ ./lab6-1
aaglushenok@fedora:~/work/arch-pc/lab06$
```

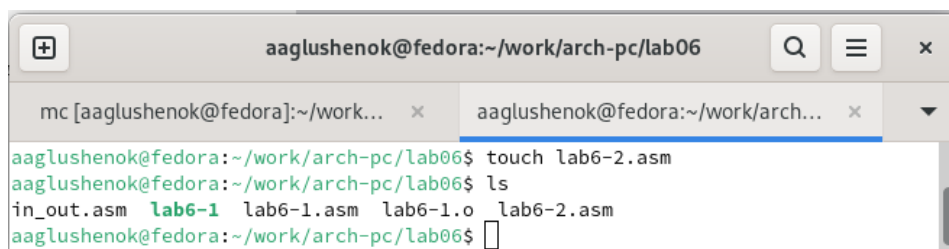
Рис. 2.5: Результат работы программы

- Преобразуйте текст программы из Листинга 6.1 с использованием функций преобразования символов в числа и обратно. Создайте файл lab6-2.asm и введите в него текст программы из листинга 6.2. Создайте исполняемый файл и запустите его.

В указанном каталоге с помощью команды touch создаем файл lab6-2.asm, вводим текст из листинга 6.2. (Программа вывода значения регистра eax). Создаем

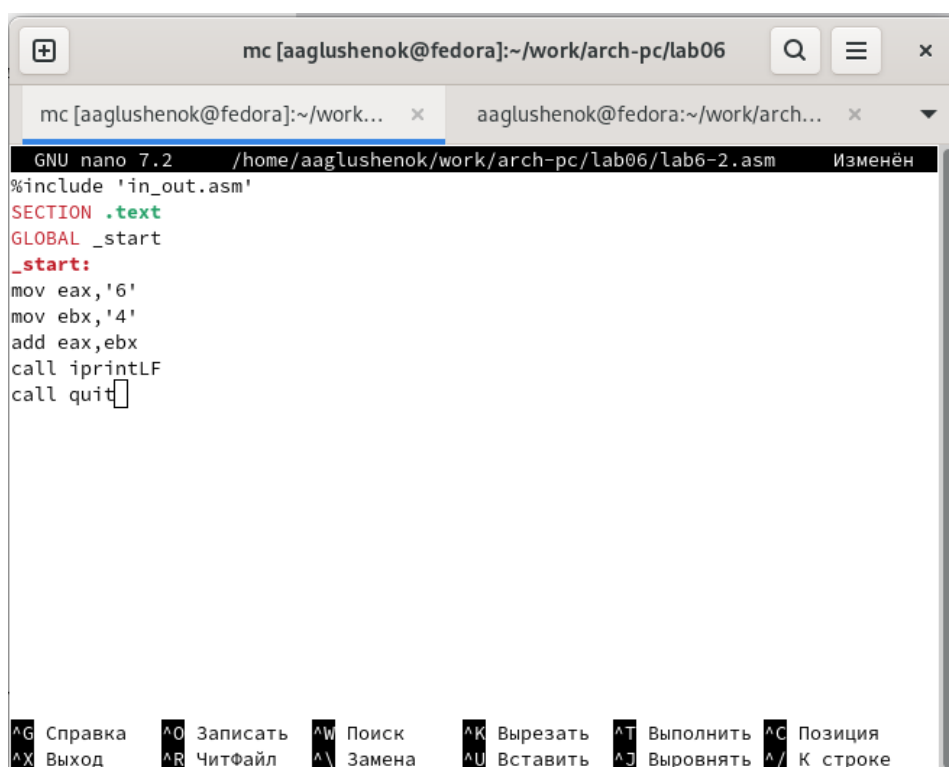


исполняемый файл и запускаем его. в результате работы программы выводится число 106, вывести которое позволяет функция `iprintLF`.



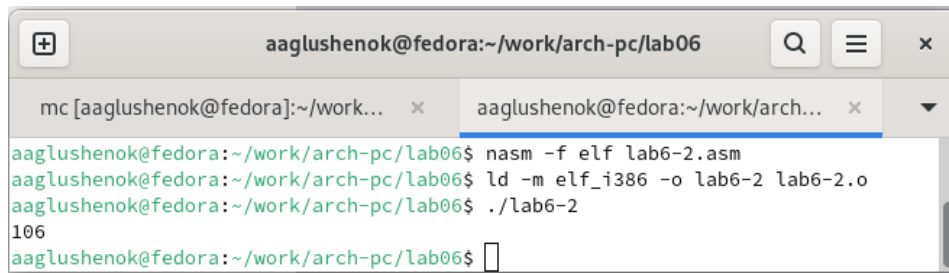
```
mc [aaglushenok@fedora]:~/work/arch-pc/lab06
aaglushenok@fedora:~/work/arch-pc/lab06$ touch lab6-2.asm
aaglushenok@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2.asm
aaglushenok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.6: Создание файла для работы



```
GNU nano 7.2 /home/aaglushenok/work/arch-pc/lab06/lab6-2.asm  Изменён
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 2.7: Программа вывода значения регистра `eax`

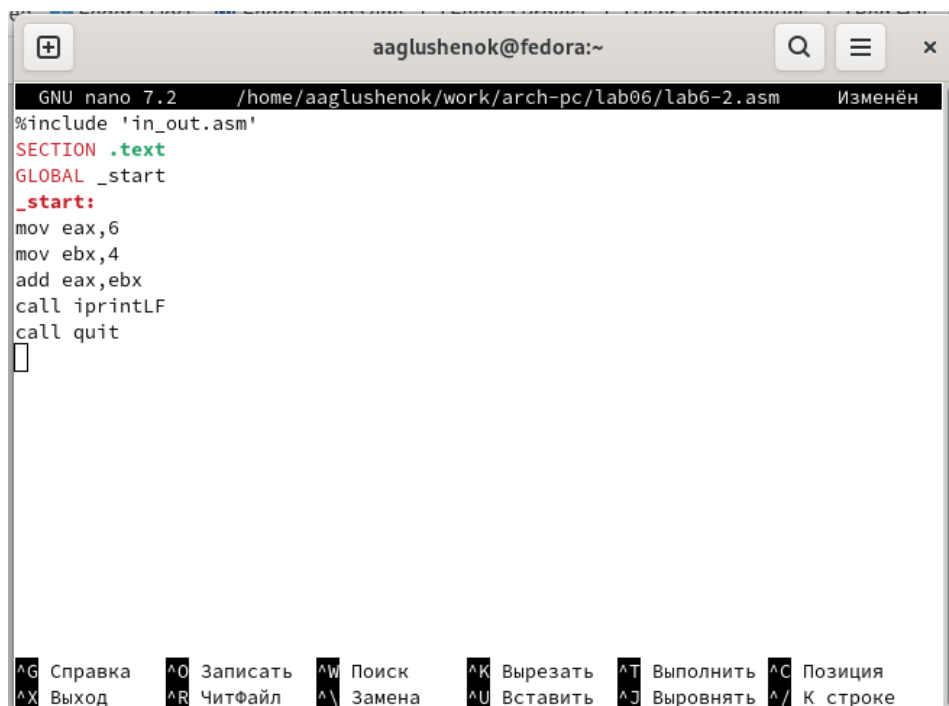


```
aaglushmanok@fedora:~/work/arch-pc/lab06
mc [aaglushmanok@fedora]:~/work... x aaglushmanok@fedora:~/work/arch... x
aaglushmanok@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
aaglushmanok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aaglushmanok@fedora:~/work/arch-pc/lab06$ ./lab6-2
106
aaglushmanok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.8: Результат работы программы

5. Аналогично предыдущему примеру измените символы на числа. Создайте исполняемый файл и запустите его. Замените функцию `iprintLF` на `iprint`. Создайте исполняемый файл и запустите его.

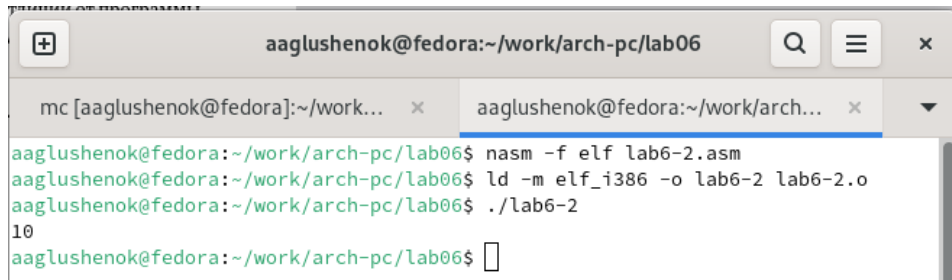
Заменяем символы на числа, убирая кавычки возле них. Создаем исполняемый файл, и в результате работы программы получаем число 10, то есть верный ответ на исходную задачу (4+6). В результате замены функции `iprintLF` на `iprint` мы так же получаем число 10, но уже переноса строки.



```
GNU nano 7.2 /home/aaglushmanok/work/arch-pc/lab06/lab6-2.asm Изменён
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit

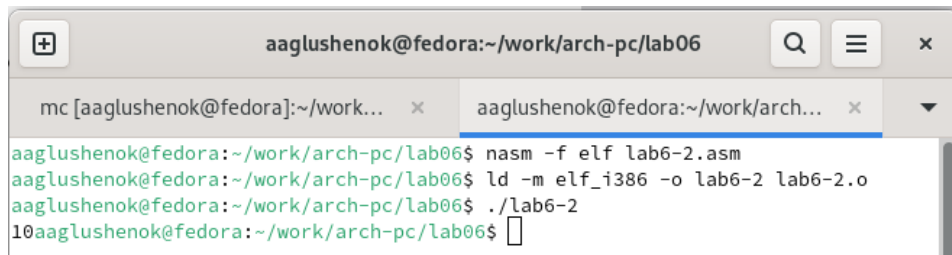
```

Рис. 2.9: Замена символов на числа

A terminal window titled 'aaglushenok@fedora:~/work/arch-pc/lab06'. It shows the following commands and output:

```
aaglushenok@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
aaglushenok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aaglushenok@fedora:~/work/arch-pc/lab06$ ./lab6-2
10
aaglushenok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.10: Результат программы с `iprintLF`

A terminal window titled 'aaglushenok@fedora:~/work/arch-pc/lab06'. It shows the following commands and output:

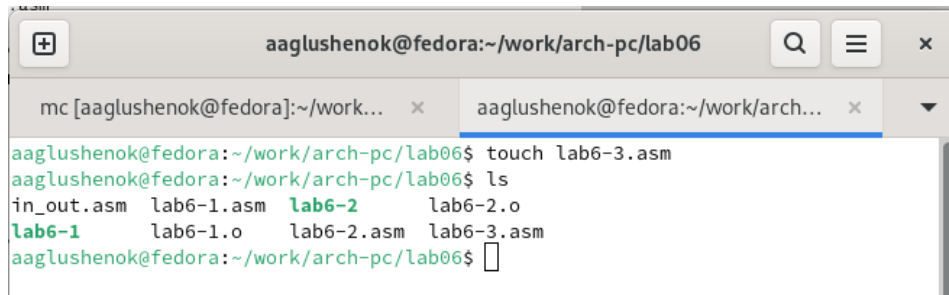
```
aaglushenok@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
aaglushenok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
aaglushenok@fedora:~/work/arch-pc/lab06$ ./lab6-2
10aaglushenok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.11: Результат программы с `iprint`

## 2.2 Задание 2

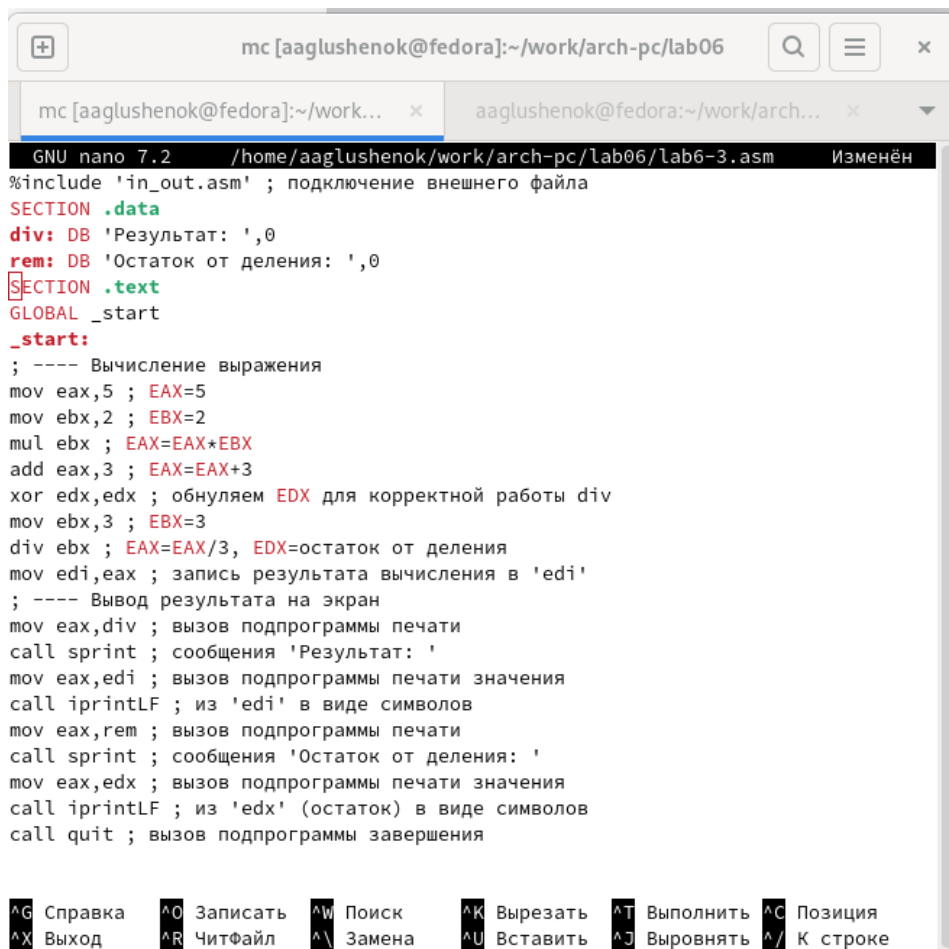
6. Создайте файл `lab6-3.asm`, введите в него текст из листинга 6.3. Создайте исполняемый файл и запустите его. Измените текст программы для вычисления выражения  $f(x) = (4 \cdot 6 + 2) / 5$ . Создайте исполняемый файл и проверьте его работу.

С помощью команды `touch` создаем указанный файл, вводим в него текст листинга 6.3. (Программа вычисления выражения  $f(x) = (5 \cdot 2 + 3) / 3$ ). Создаем исполняемый файл, запускаем его. В результате работы программы получаем частное и остаток заданного выражения. Вносим изменения в программу, и в ответе, аналогично, получаем частное и остаток от второго выражения.

A terminal window titled 'aaglushenok@fedora:~/work/arch-pc/lab06'. It shows the execution of 'touch lab6-3.asm' and 'ls'. The 'ls' command lists files: 'in\_out.asm', 'lab6-1.asm', 'lab6-2', 'lab6-2.o', 'lab6-1', 'lab6-1.o', 'lab6-2.asm', and 'lab6-3.asm'.

```
aaglushenok@fedora:~/work/arch-pc/lab06$ touch lab6-3.asm
aaglushenok@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm  lab6-2      lab6-2.o
lab6-1      lab6-1.o    lab6-2.asm  lab6-3.asm
aaglushenok@fedora:~/work/arch-pc/lab06$
```

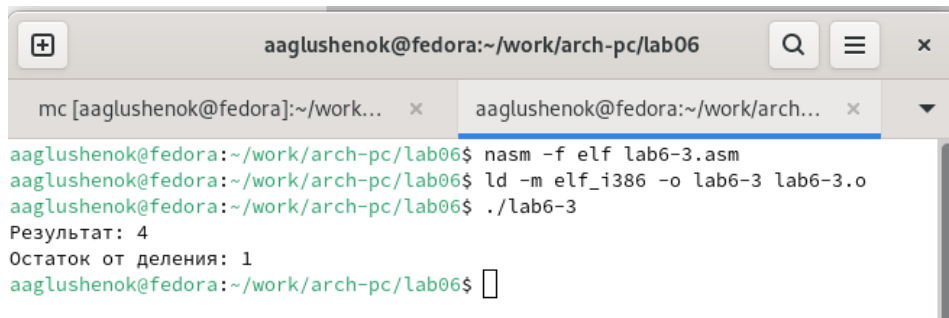
Рис. 2.12: Создание файла

A Nano 7.2 editor window titled 'GNU nano 7.2 /home/aaglushenok/work/arch-pc/lab06/lab6-3.asm'. It contains assembly code for calculating (5\*2+3)/3. Comments in Russian explain each step: including a file, defining data, calculating the expression, and printing the result and remainder. The bottom status bar shows keyboard shortcuts for various editor functions.

```
GNU nano 7.2 /home/aaglushenok/work/arch-pc/lab06/lab6-3.asm  Изменён
%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: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

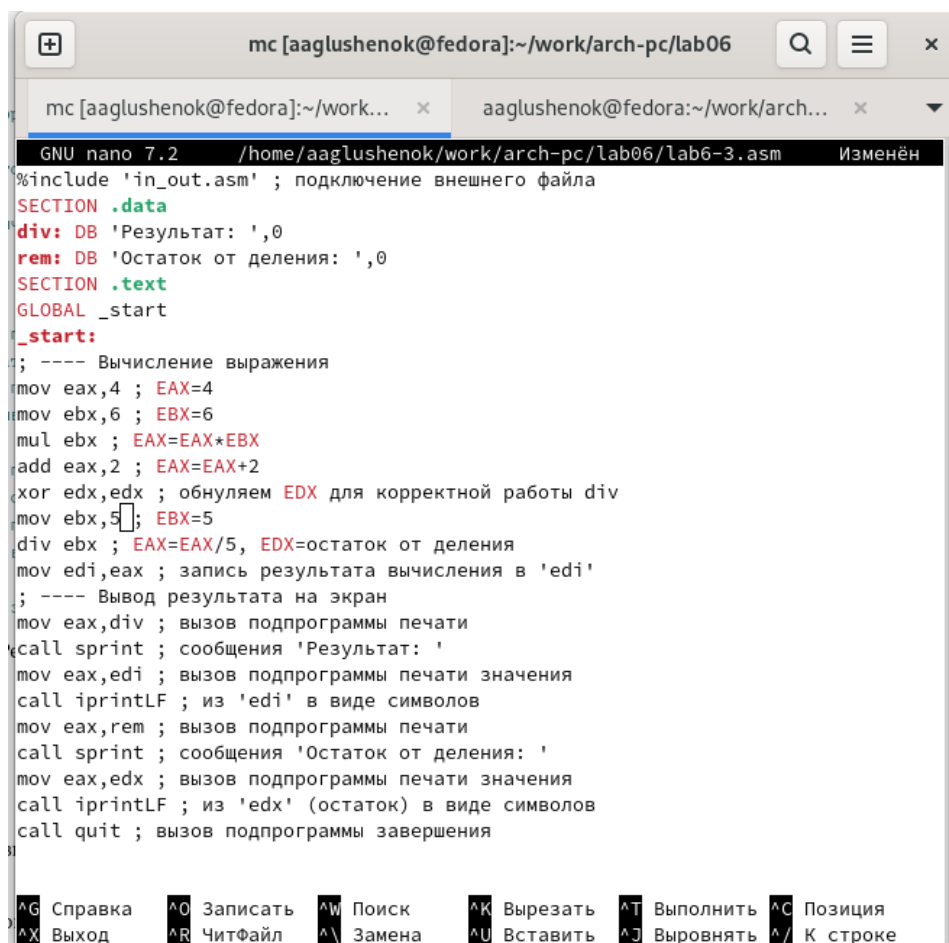
^G Справка      ^O Записать     ^W Поиск       ^K Вырезать    ^T Выполнить   ^C Позиция
^X Выход        ^R ЧитФайл     ^_ Замена      ^U Вставить    ^J Выровнять   ^/ К строке
```

Рис. 2.13: Программа вычисления выражения  $f(x)=(5*2+3)/3$



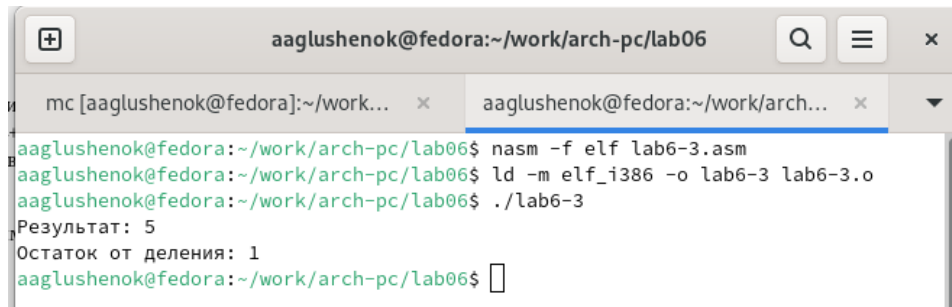
```
aaglushmanok@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
aaglushmanok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
aaglushmanok@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
aaglushmanok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.14: Результат работы программы



```
GNU nano 7.2 /home/aaglushmanok/work/arch-pc/lab06/lab6-3.asm Изменён
%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 iprintLF ; 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 iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 2.15: Программа вычисления выражения  $f(x)=(4*6+2)/5$

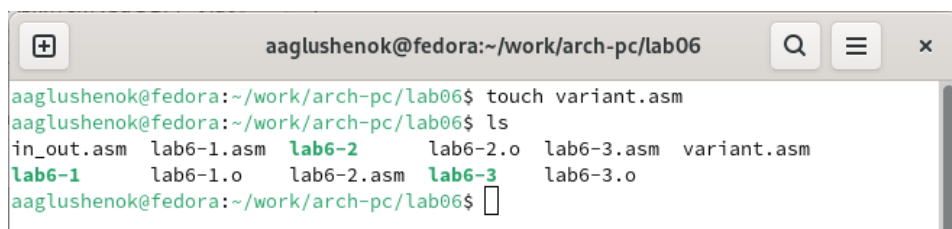


```
aaglushmanok@fedora:~/work/arch-pc/lab06
mc [aaglushmanok@fedora]:~/work... x aaglushmanok@fedora:~/work/arch... x
aaglushmanok@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
aaglushmanok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
aaglushmanok@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
aaglushmanok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.16: Результат работы программы

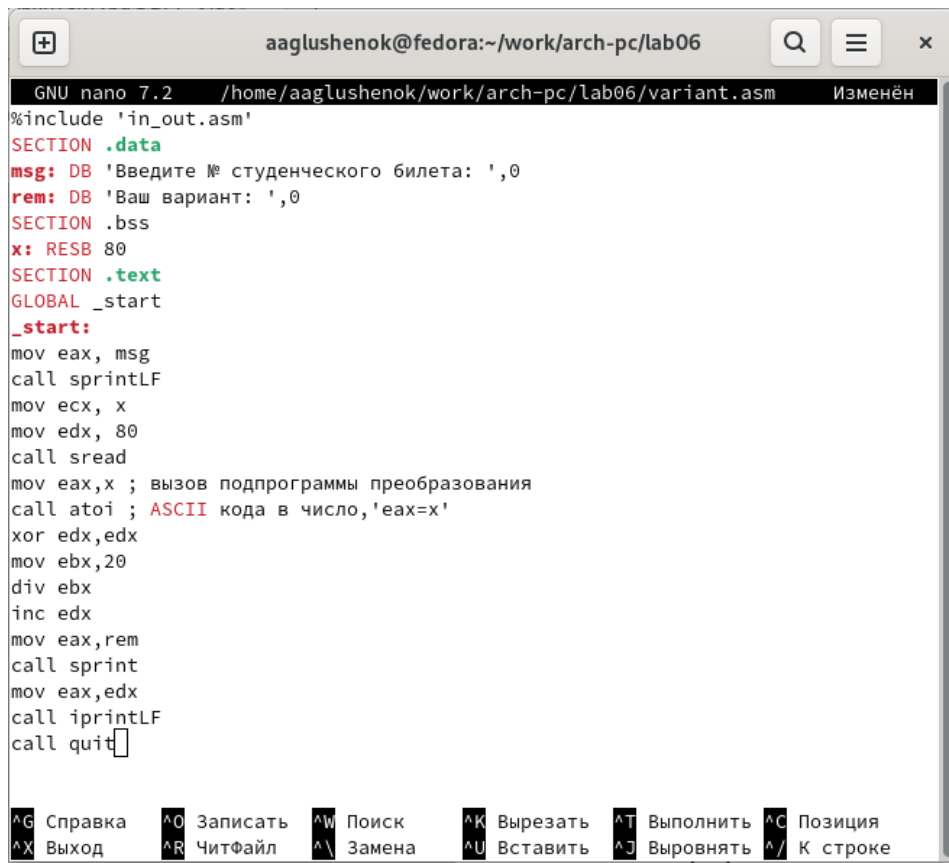
7. Рассмотрите программу вычисления варианта задания по номеру студенческого билета (запрос номера билета, вычисление варианта по формуле, вывод номера варианта). Создайте файл `variant.asm`, введите в него текст листинга 6.4. Создайте исполняемый файл и запустите его. Проверьте результат работы программы вычислив номер варианта.

Для реализации данной программы создаем файл `variant.asm` и вводим в него текст листинга 6.2. (Программа вычисления варианта задания по номеру студенческого билета). Создаем исполняемый файл. В результате его работы получаем запрос на ввод номера студ билета, и после ввода номера получаем цифру, соответствующую варианту. Проверяем результат работы программы с помощью калькулятора.



```
aaglushmanok@fedora:~/work/arch-pc/lab06
aaglushmanok@fedora:~/work/arch-pc/lab06$ touch variant.asm
aaglushmanok@fedora:~/work/arch-pc/lab06$ ls
in_out.asm lab6-1.asm lab6-2 lab6-2.o lab6-3.asm variant.asm
lab6-1 lab6-1.o lab6-2.asm lab6-3 lab6-3.o
aaglushmanok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.17: Создание файла для работы

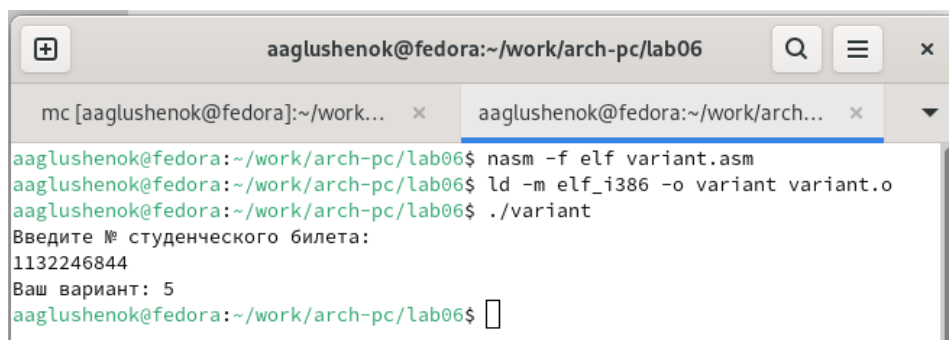


The screenshot shows a terminal window with the nano text editor open. The title bar indicates the user is 'aaglushenok' on a 'fedora' machine, in the directory '~/work/arch-pc/lab06'. The editor is editing a file named 'variant.asm'. The code is as follows:

```
GNU nano 7.2 /home/aaglushenok/work/arch-pc/lab06/variant.asm
%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 sprintLF
mov ecx, x
mov edx, 80
call sread
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 sprint
mov eax, edx
call iprintLF
call quit
```

At the bottom of the editor, there is a status bar with keyboard shortcuts: ^G Справка, ^O Записать, ^W Поиск, ^K Вырезать, ^T Выполнить, ^C Позиция, ^X Выход, ^R ЧитФайл, ^\_ Замена, ^U Вставить, ^J Выводить, ^/ К строке.

Рис. 2.18: Программа вычисления варианта по номеру студ билета



The screenshot shows a terminal window with the same user and directory as the previous image. It shows the execution of the assembly program. The commands and their outputs are:

```
aaglushenok@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
aaglushenok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
aaglushenok@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246844
Ваш вариант: 5
aaglushenok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.19: Результат работы программы

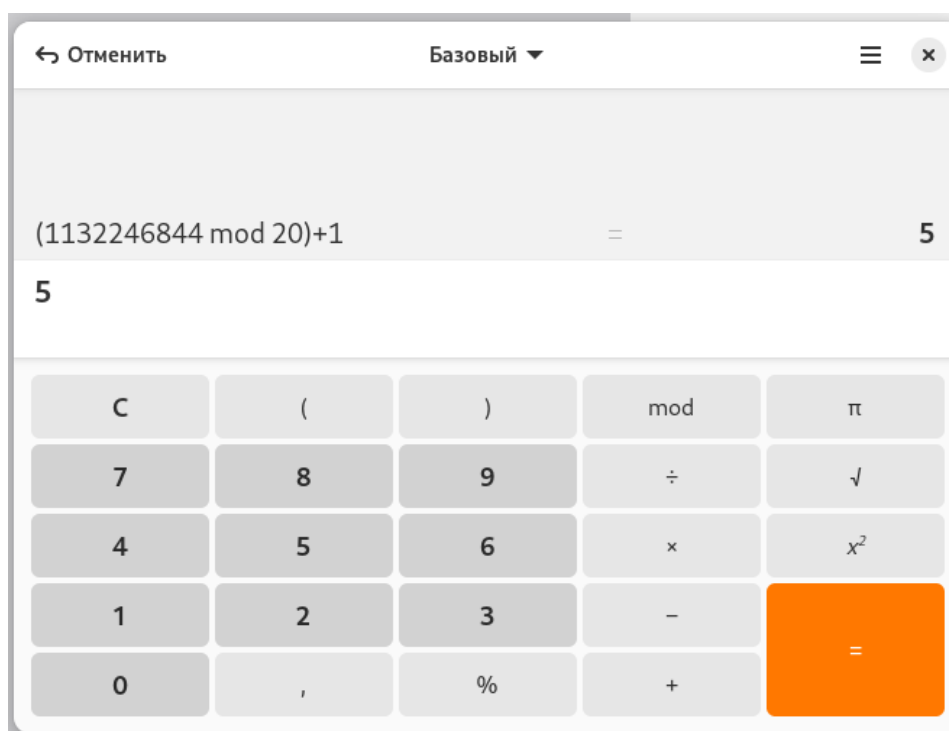


Рис. 2.20: Проверка результата программы

## 2.3 Ответы на вопросы

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения 'Ваш вариант:'? ОТВЕТ: За вывод на экран сообщения 'Ваш вариант:' отвечает строка "mov eax,rem" и строка "call sprint".
2. Для чего используются следующие инструкции? mov ecx, x mov edx, 80 call sread ОТВЕТ: Иструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре ecx, а максимальное количество считываемых символов строки в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки.
3. Для чего используется инструкция "call atoi"? ОТВЕТ: Инструкция "call atoi" используется для преобразования строки в целое число. Она принимает адрес строки в регистре eax и возвращает полученное число в нем же.



4. Какие строки листинга 6.4 отвечают за вычисление варианта? ОТВЕТ: 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. В какой регистр записывается остаток от деления при выполнении инструкции `"div ebx"`? ОТВЕТ: Остаток от деления записывается в регистр `edx`.
6. Для чего используется инструкция `"inc edx"`? ОТВЕТ: Инструкция `"inc edx"` используется для увеличения значения в регистре `edx` на 1. В данном случае, она увеличивает остаток от деления на 1.
7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычисления? ОТВЕТ: 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.

## 2.4 Задания для самостоятельной работы

Написать программу вычисления выражения  $y = f(x)$ , где  $f(x) = (9x - 8)/8$  (вариант 5), проверить правильность работы программы для  $x=8$  и  $x=64$ .

Аналогично предшествующим программам, создаем новую программу для вычисления значения функции. Затем создаем исполняемый файл. Программа запрашивает у пользователя ввести значение  $x$ , дает возможность ввода, после чего вычисляет и выводит значение функции при заданном пользователем значении  $x$ .

GNU nano 7.2

```

#include 'in_out.asm'
SECTION .data
msg: DB 'Введите x:', 0
div: DB 'Результат: ', 0
SECTION .bss
rez: RESB 80
x RESB 80
SECTION .text
global _start
_start:
; Запрос ввода значения x
mov eax, msg
call sprintLF

; Чтение значения x
mov ecx, x
mov edx, 80
call sread

; Преобразование строки в целое число
mov eax, x
call atoi

; Умножаем x на 9
mov ebx, eax      ; сохраняем x в ebx
mov ecx, 9
imul eax, ecx     ; eax = 9 * x

; Вычитаем 8
sub eax, 8        ; eax = 9 * x - 8

; Делим на 8
mov ebx, 8
cdq               ; расширяем eax в edx:eax для деления
idiv ebx          ; eax = (9 * x - 8) / 8

; Сохраняем результат в `rez`
mov [rez], eax

; Выводим результат
mov eax, div
call sprint
mov eax, [rez]
call iprintLF

; Завершаем программу
call quit

```

^G Справка  
^X Выход

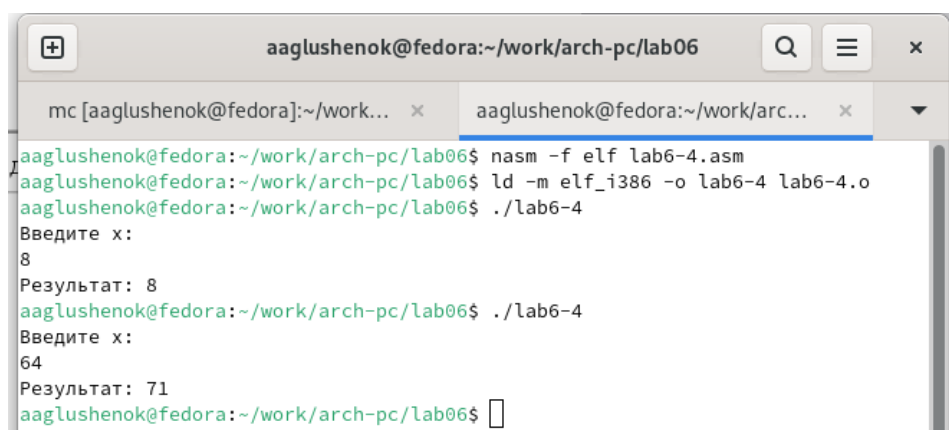
^O Записать  
^R ЧитФайл

^W Поиск  
^\_ Замена

^K Вырезать  
^U Вставить

^T Выполнить  
^J Выровнять

Рис. 2.21: Создание программы

A terminal window titled 'aaglushenok@fedora:~/work/arch-pc/lab06'. It shows the compilation of an assembly file 'lab6-4.asm' into an object file 'lab6-4.o' using 'nasm', followed by linking into an executable 'lab6-4' using 'ld'. The user then runs the program twice, providing input '8' and '64', and the program outputs 'Результат: 8' and 'Результат: 71' respectively. The terminal window has a standard Linux desktop interface with a title bar and window controls.

```
aaglushenok@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
aaglushenok@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
aaglushenok@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
8
Результат: 8
aaglushenok@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
64
Результат: 71
aaglushenok@fedora:~/work/arch-pc/lab06$
```

Рис. 2.22: Проверка работы программы

## **3 Выводы**

В ходе выполнения лабораторной работы мне удалось освоить арифметические инструкции языка ассемблера NASM.