

Отчёт по лабораторной работе № 7

НКАбд-04-22

Безнощук Владимир Юрьевич

Содержание

Цель работы	1
Задание.....	1
Порядок выполнения лабораторной работы.....	1
Символьные и численные данные в NASM	1
Выполнение арифметических операций в NASM	6
Задание для самостоятельной работы.....	11
Выводы.....	13

Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

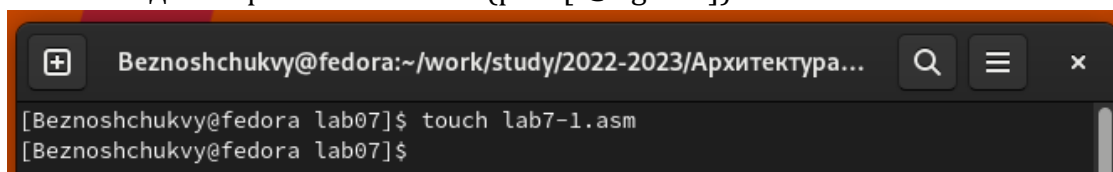
Задание

1. выполнить работу с символьными и численными данными в NASM
2. Отработать на практике арифметические операции в NASM
3. Написать программу вычисления выражения с входными данными

Порядок выполнения лабораторной работы

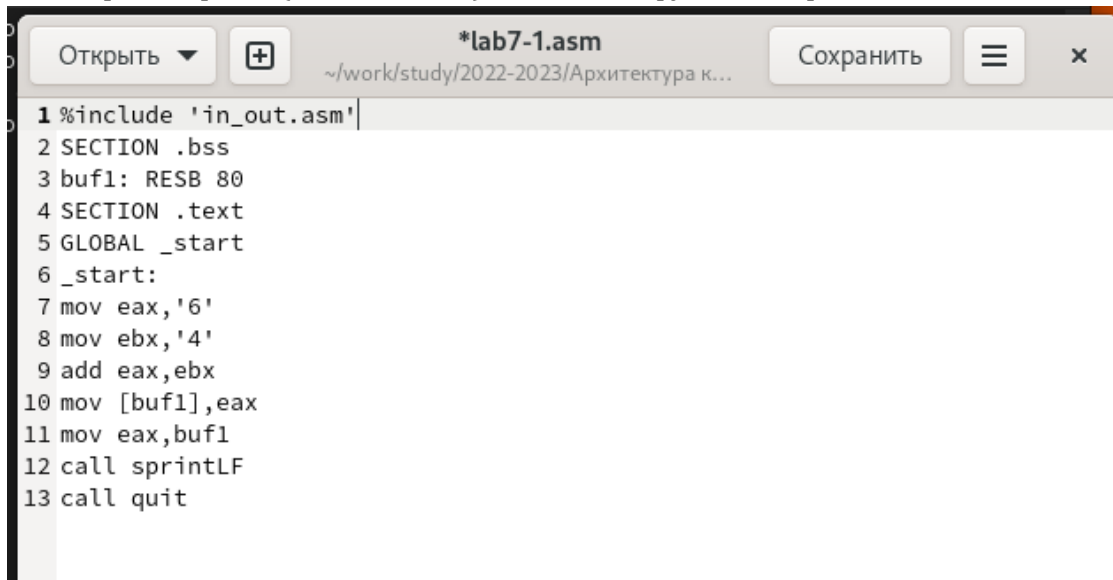
Символьные и численные данные в NASM

1. Создали каталог для программ лабораторной работы № 7, перешли в него и создайте файл lab7-1.asm. (рис. [-@fig:001])

A screenshot of a terminal window with a dark background. The title bar at the top shows the user 'Beznoshchukvy@fedora' and the directory '~/work/study/2022-2023/Архитектура...'. The terminal text shows the command 'touch lab7-1.asm' being executed in the 'lab07' directory, followed by a new prompt line.

Файл lab7-1.asm

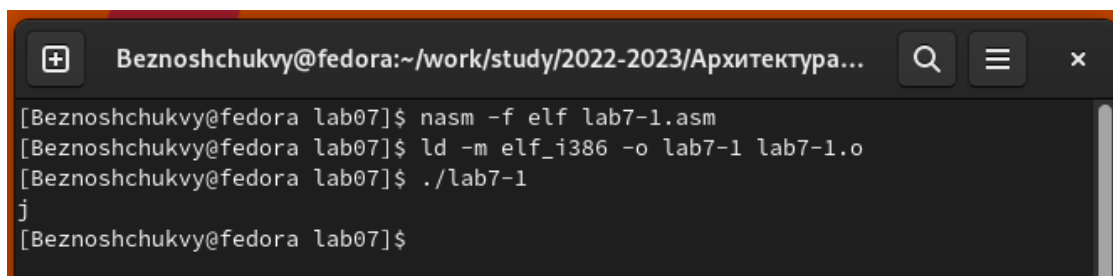
2. Рассмотрели примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр еах. Ввели в файл lab7-1.asm текст программы из листинга 7.1. (рис. [-@fig:002]) В данной программе в регистр еах записывается символ 6 (mov еах,'6'), в регистр еbх символ 4 (mov еbх,'4'). Далее к значению в регистре еах прибавляется значение регистра еbх (add еах,еbх, результат сложения запишется в регистр еах). Далее выводится результат. Так как для работы функции sprintLF в регистр еах должен быть записан адрес, необходимо использовать дополнительную переменную. Для этого записали значение регистра еах в переменную buf1 (mov [buf1],еах), а затем записали адрес переменной buf1 в регистр еах (mov еах,buf1) и вызвали функцию sprintLF.



```
*lab7-1.asm
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov еах,'6'
8 mov еbх,'4'
9 add еах,еbх
10 mov [buf1],еах
11 mov еах,buf1
12 call sprintLF
13 call quit
```

Текст программы из листинга 7.1

Создали исполняемый файл и запустили его. (рис. [-@fig:003])



```
Beznoshchukvy@fedora:~/work/study/2022-2023/Архитектура...
[Beznoshchukvy@fedora lab07]$ nasm -f elf lab7-1.asm
[Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[Beznoshchukvy@fedora lab07]$ ./lab7-1
j
[Beznoshchukvy@fedora lab07]$
```

Исполняемый файл

3. Далее изменили текст программы и вместо символов, записали в регистры числа. (рис. [-@fig:004])

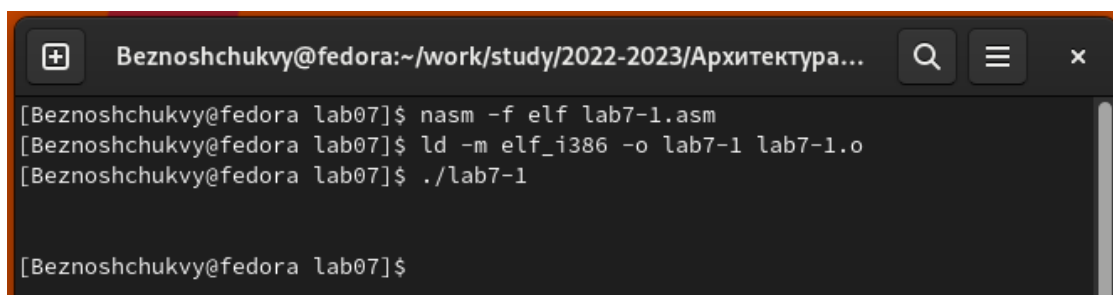


```
Открыть ▾ + • lab7-1.asm
~/work/study/2022-2023/Архитекту...tudy_2022-2023_arh-pc/labs/lab07

%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
```

Текст программы

Создали исполняемый файл и запустили его. (рис. [-@fig:005])



```
+ Beznoshchukvy@fedora:~/work/study/2022-2023/Архитектура... 🔍 ≡ ×

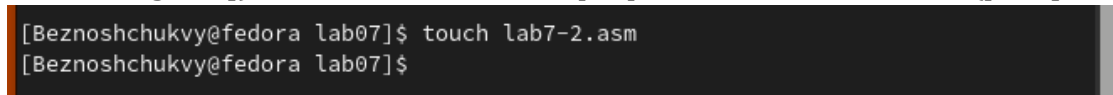
[Beznoshchukvy@fedora lab07]$ nasm -f elf lab7-1.asm
[Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[Beznoshchukvy@fedora lab07]$ ./lab7-1

[Beznoshchukvy@fedora lab07]$
```

Исполняемый файл

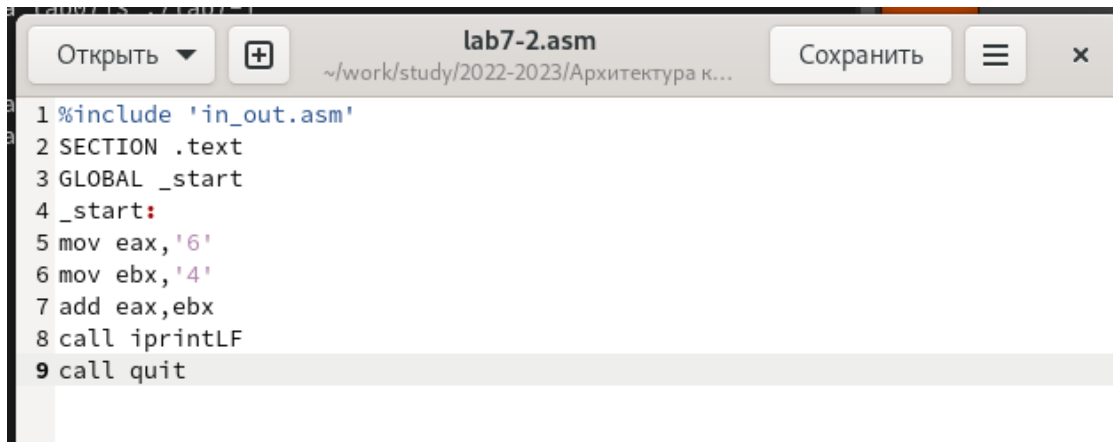
Как и в предыдущем случае при исполнении программы мы не получим число 10. В данном случае выводится символ с кодом 10. Пользуясь таблицей ASCII определили, что код 10 соответствует символ /n. Это символ перевода строки, он не отображается.

4. Как отмечалось выше, для работы с числами в файле in_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразовали текст программы из Листинга 7.1 с использованием этих функций. Создали файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 (рис. [-@fig:006]) и ввели в него текст программы из листинга 7.2. (рис. [-@fig:007])



```
[Beznoshchukvy@fedora lab07]$ touch lab7-2.asm
[Beznoshchukvy@fedora lab07]$
```

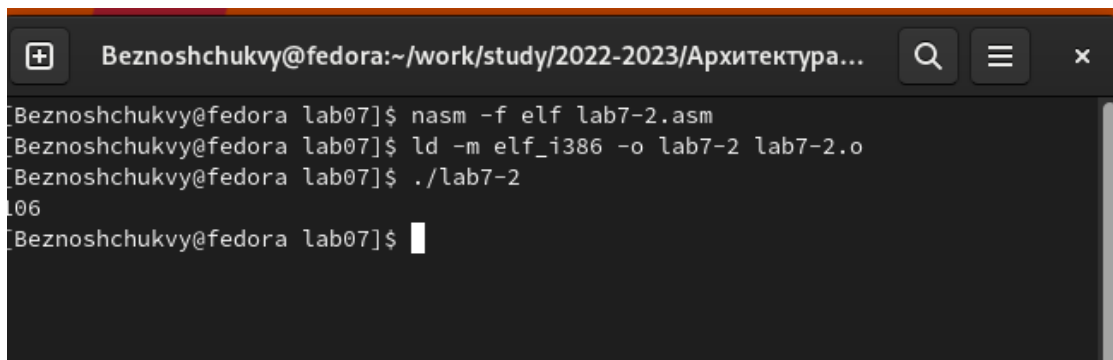
файл lab7-2.asm

A screenshot of a text editor window titled 'lab7-2.asm'. The window has a menu bar with 'Открыть' (Open), a plus icon, and 'Сохранить' (Save). Below the menu bar is a path: '~/work/study/2022-2023/Архитектура к...'. The main text area contains the following assembly code:

```
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
```

Текст программы из листинга 7.2

Создали исполняемый файл и запустили его. (рис. [-@fig:008])

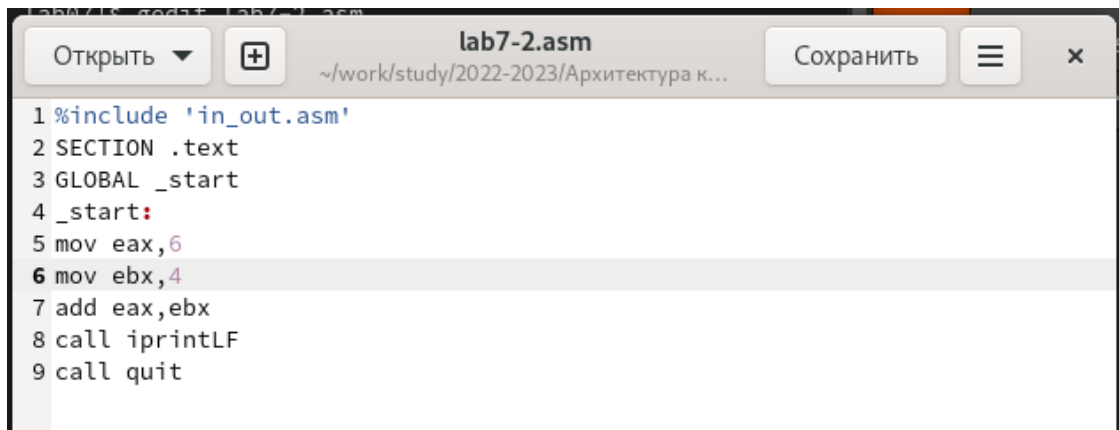
A screenshot of a terminal window. The title bar shows the user 'Beznoshchukvy' at 'fedora' in the directory '~/work/study/2022-2023/Архитектура...'. The terminal contains the following commands and output:

```
Beznoshchukvy@fedora lab07]$ nasm -f elf lab7-2.asm
Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
Beznoshchukvy@fedora lab07]$ ./lab7-2
106
Beznoshchukvy@fedora lab07]$
```

Исполняемый файл

В результате работы программы мы получили 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.

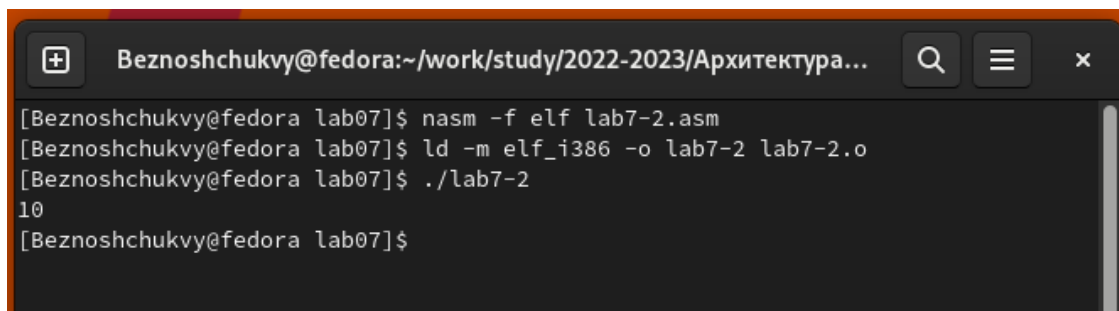
5. Аналогично предыдущему примеру изменим символы на числа. (рис. [-@fig:009])



```
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
```

Изменённый код

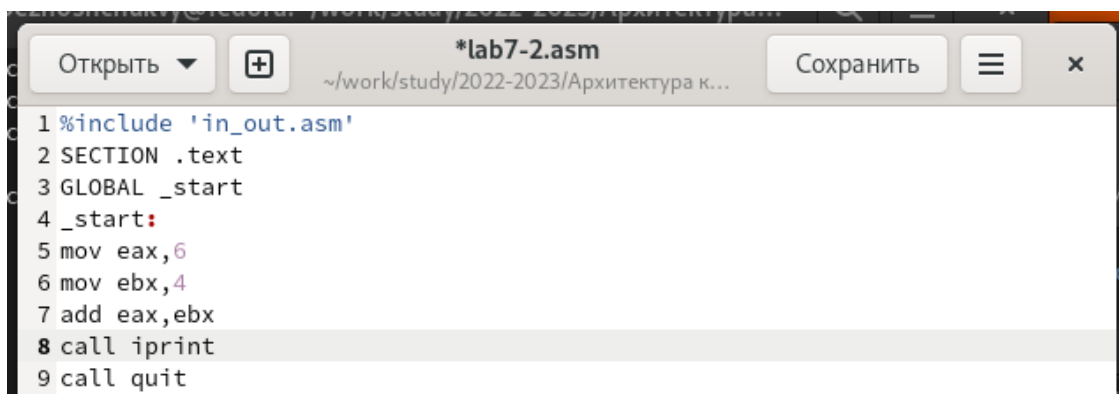
Создайте исполняемый файл и запустите его. В результате при исполнении программы получили 10. (рис. [-@fig:010])



```
[Beznoshchukvy@fedora lab07]$ nasm -f elf lab7-2.asm
[Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[Beznoshchukvy@fedora lab07]$ ./lab7-2
10
[Beznoshchukvy@fedora lab07]$
```

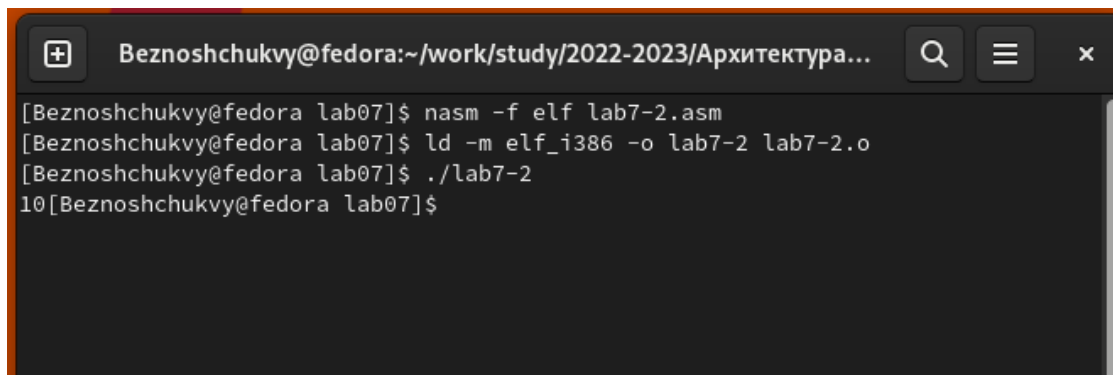
Результат программы

Заменяли функцию iprintLF на iprint. (рис. [-@fig:011]) Создайте исполняемый файл и запустите его. (рис. [-@fig:012]) Вывод функций iprintLF и iprint отличается наличием перевода строки после вывода?



```
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
```

iprintLF на iprint

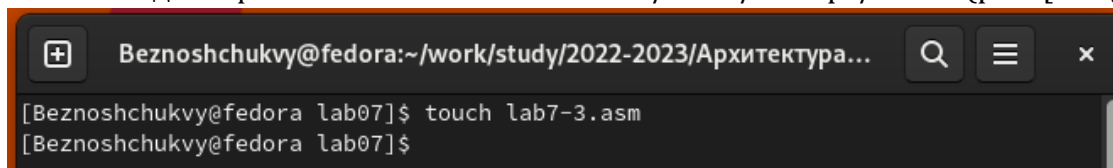


```
Beznoshchukvy@fedora:~/work/study/2022-2023/Архитектура...  
[Beznoshchukvy@fedora lab07]$ nasm -f elf lab7-2.asm  
[Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o  
[Beznoshchukvy@fedora lab07]$ ./lab7-2  
10[Beznoshchukvy@fedora lab07]$
```

Работа исполняемого файла

Выполнение арифметических операций в NASM

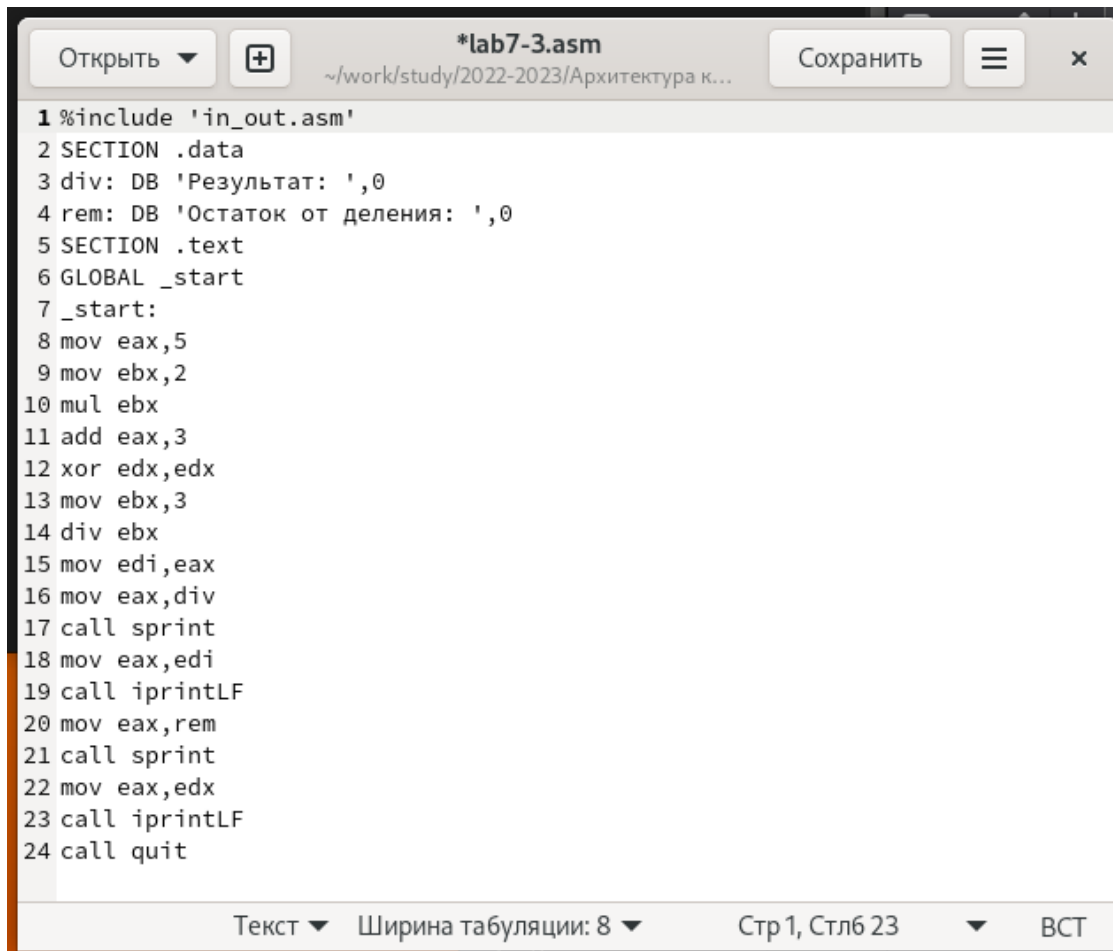
6. В качестве примера выполнения арифметических операций в NASM привели программу вычисления арифметического выражения $\mathbb{Z}(\mathbb{Z}) = (5 * 2 + 3)/3$. Создали файл lab7-3.asm в каталоге ~/work/arch-рс/lab07. (рис. [-@fig:013])



```
Beznoshchukvy@fedora:~/work/study/2022-2023/Архитектура...  
[Beznoshchukvy@fedora lab07]$ touch lab7-3.asm  
[Beznoshchukvy@fedora lab07]$
```

Файл lab7-3.asm

Внимательно изучили текст программы из листинга 7.3 и ввели в lab7-3.asm. (рис. [-@fig:014])



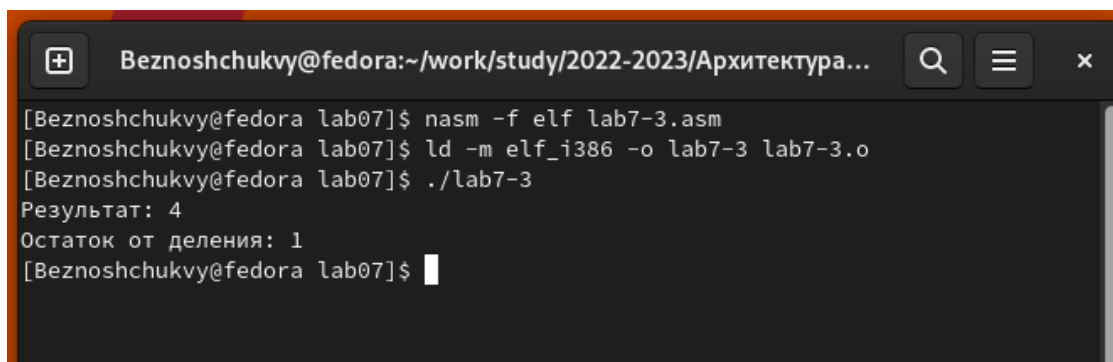
The screenshot shows a text editor window titled '*lab7-3.asm' with a menu bar containing 'Открыть', '+', '~/.work/study/2022-2023/Архитектура к...', 'Сохранить', and window controls. The code is as follows:

```
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 mov eax,5
9 mov ebx,2
10 mul ebx
11 add eax,3
12 xor edx,edx
13 mov ebx,3
14 div ebx
15 mov edi,eax
16 mov eax,div
17 call sprint
18 mov eax,edi
19 call iprintLF
20 mov eax,rem
21 call sprint
22 mov eax,edx
23 call iprintLF
24 call quit
```

The status bar at the bottom indicates 'Текст', 'Ширина табуляции: 8', 'Стр 1, Стлб 23', and 'ВСТ'.

Текст программы из листинга 7.3

Создали исполняемый файл и запустили его. Получили следующий результат. (рис. [-@fig:015])

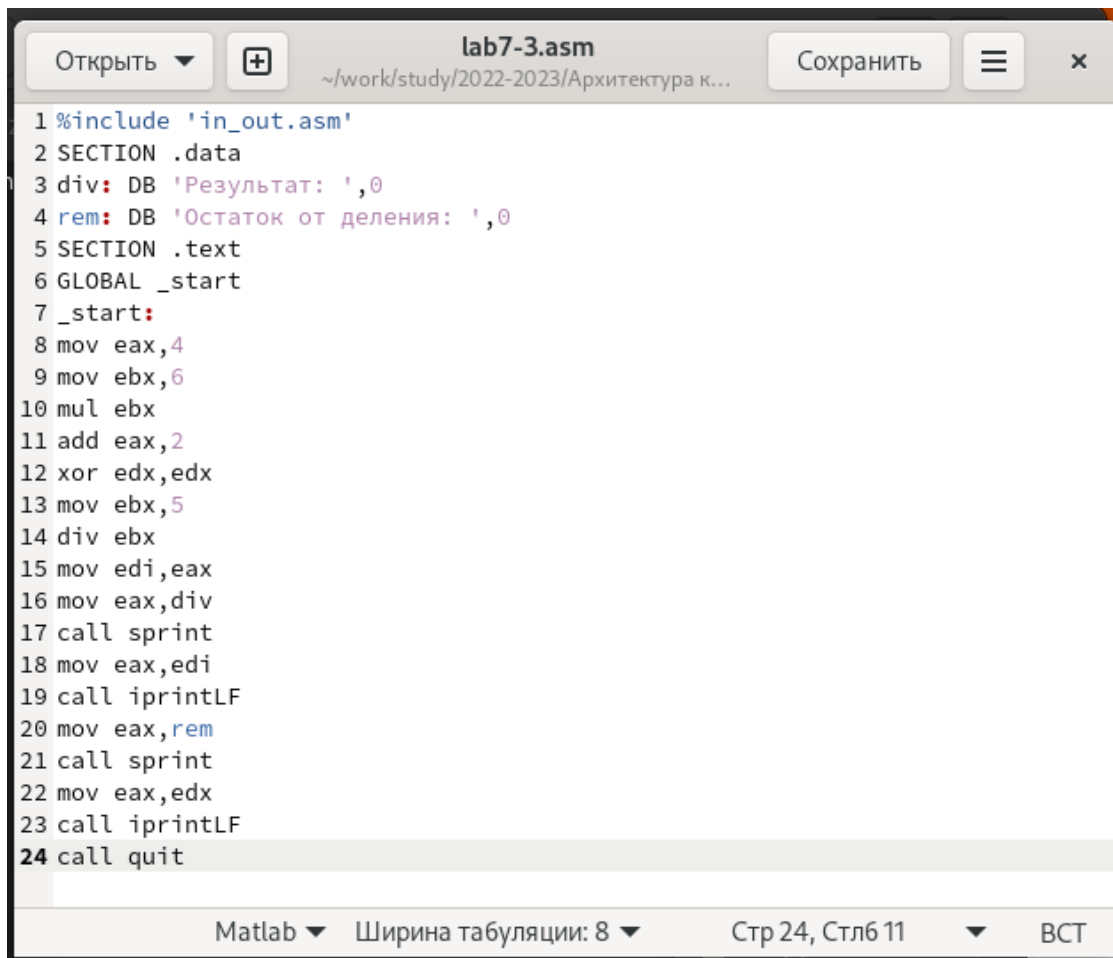


The screenshot shows a terminal window with the following commands and output:

```
[Beznoshchukvy@fedora lab07]$ nasm -f elf lab7-3.asm
[Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[Beznoshchukvy@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[Beznoshchukvy@fedora lab07]$
```

Исполняемый файл

Изменили текст программы для вычисления выражения $\frac{4 * 6 + 2}{5}$. (рис. [-@fig:016])



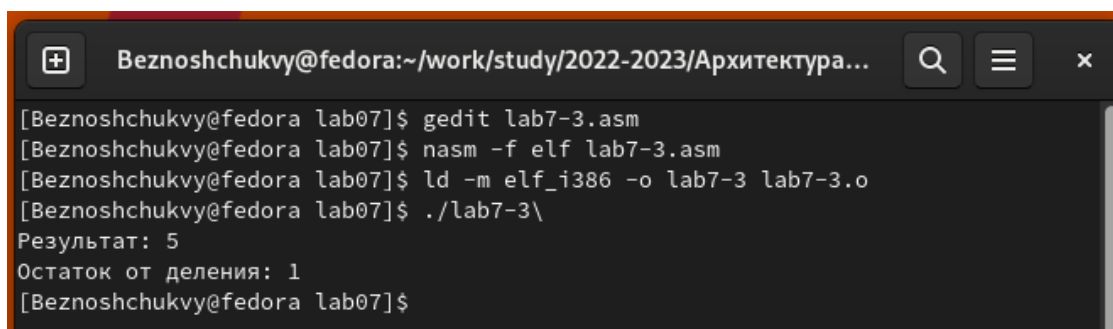
The screenshot shows a text editor window titled "lab7-3.asm" with a file path of "~/work/study/2022-2023/Архитектура к...". The code is as follows:

```
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 mov eax,4
9 mov ebx,6
10 mul ebx
11 add eax,2
12 xor edx,edx
13 mov ebx,5
14 div ebx
15 mov edi,eax
16 mov eax,div
17 call sprint
18 mov eax,edi
19 call iprintLF
20 mov eax,rem
21 call sprint
22 mov eax,edx
23 call iprintLF
24 call quit
```

The status bar at the bottom indicates "Matlab", "Ширина табуляции: 8", "Стр 24, Стлб 11", and "ВСТ".

Изменённый текст программы

Создали исполняемый файл и проверили его работу. (рис. [-@fig:017])



The screenshot shows a terminal window with the following commands and output:

```
[Beznoshchukvy@fedora lab07]$ gedit lab7-3.asm
[Beznoshchukvy@fedora lab07]$ nasm -f elf lab7-3.asm
[Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[Beznoshchukvy@fedora lab07]$ ./lab7-3\
Результат: 5
Остаток от деления: 1
[Beznoshchukvy@fedora lab07]$
```

Исполняемый файл

7. В качестве другого примера рассмотрели программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:

- вывести запрос на введение № студенческого билета
- вычислить номер варианта по формуле: $(\text{номер билета} \bmod 20) + 1$, где номер билета – номер студенческого билета (В данном случае $\text{номер билета} \bmod 20$ – это остаток от деления номер билета на 20).
- вывести на экран номер варианта.

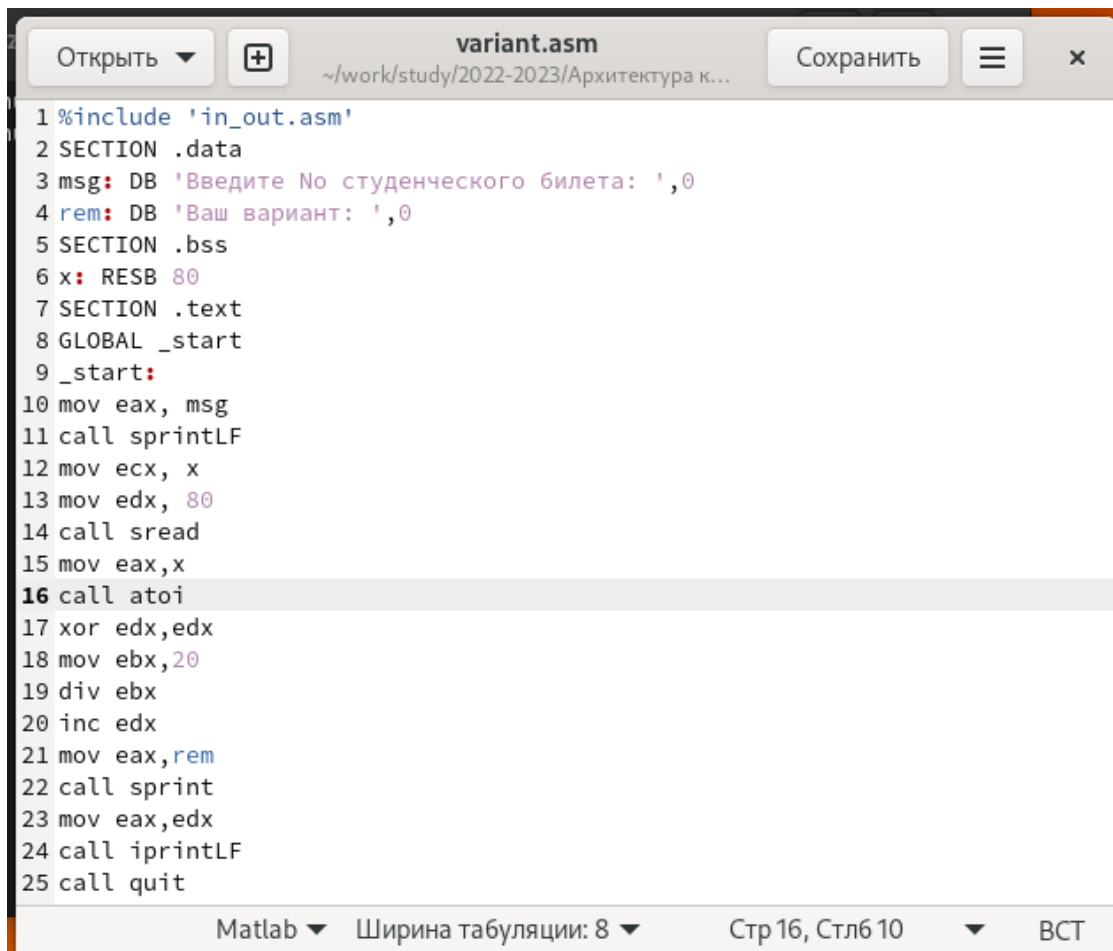
В данном случае число, над которым необходимо проводить арифметические операции, вводится с клавиатуры. Как отмечалось выше, ввод с клавиатуры осуществляется в символьном виде и для корректной работы арифметических операций в NASM символы преобразуются в числа. Для этого использована функция `atoi` из файла `in_out.asm`.

Создали файл `variant.asm` в каталоге `~/work/arch-pc/lab07`. (рис. [-@fig:018])

A terminal window with a dark background and orange title bar. The title bar contains a plus icon, the text 'Beznoshchukvy@fedora:~/work/study/2022-2023/Архитектура...', a search icon, a menu icon, and a close icon. The terminal shows two lines of text: '[Beznoshchukvy@fedora lab07]\$ touch variant.asm' and '[Beznoshchukvy@fedora lab07]\$'.

Файл `variant.asm`

Внимательно изучили текст программы из листинга 7.4 и ввели в файл `variant.asm`. (рис. [-@fig:019])

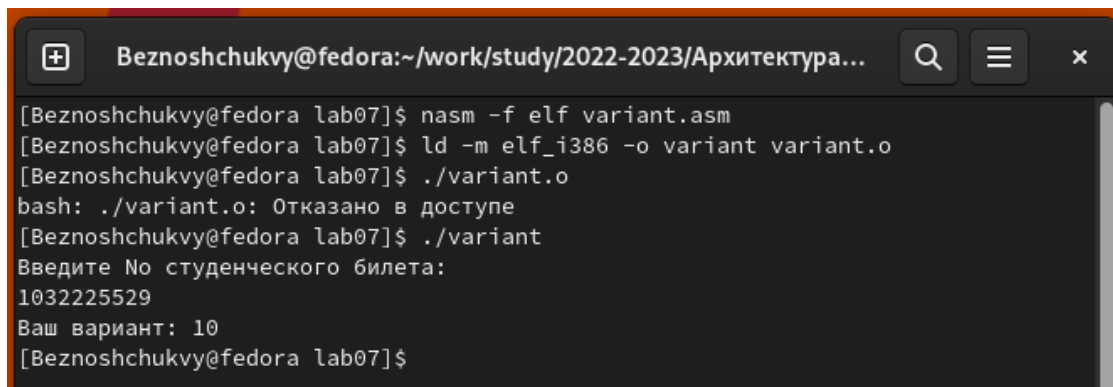


```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите No студенческого билета: ',0
4 rem: DB 'Ваш вариант: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintf
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x
16 call atoi
17 xor edx,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

Matlab ▾ Ширина табуляции: 8 ▾ Стр 16, Стлб 10 ▾ ВСТ

Текст программы из листинга 7.4

Создали исполняемый файл и запустили его. (рис. [-@fig:020]) Проверили результат работы программы вычислив номер варианта аналитически.



```
[Beznoshchukvy@fedora lab07]$ nasm -f elf variant.asm
[Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[Beznoshchukvy@fedora lab07]$ ./variant.o
bash: ./variant.o: Отказано в доступе
[Beznoshchukvy@fedora lab07]$ ./variant
Введите No студенческого билета:
1032225529
Ваш вариант: 10
[Beznoshchukvy@fedora lab07]$
```

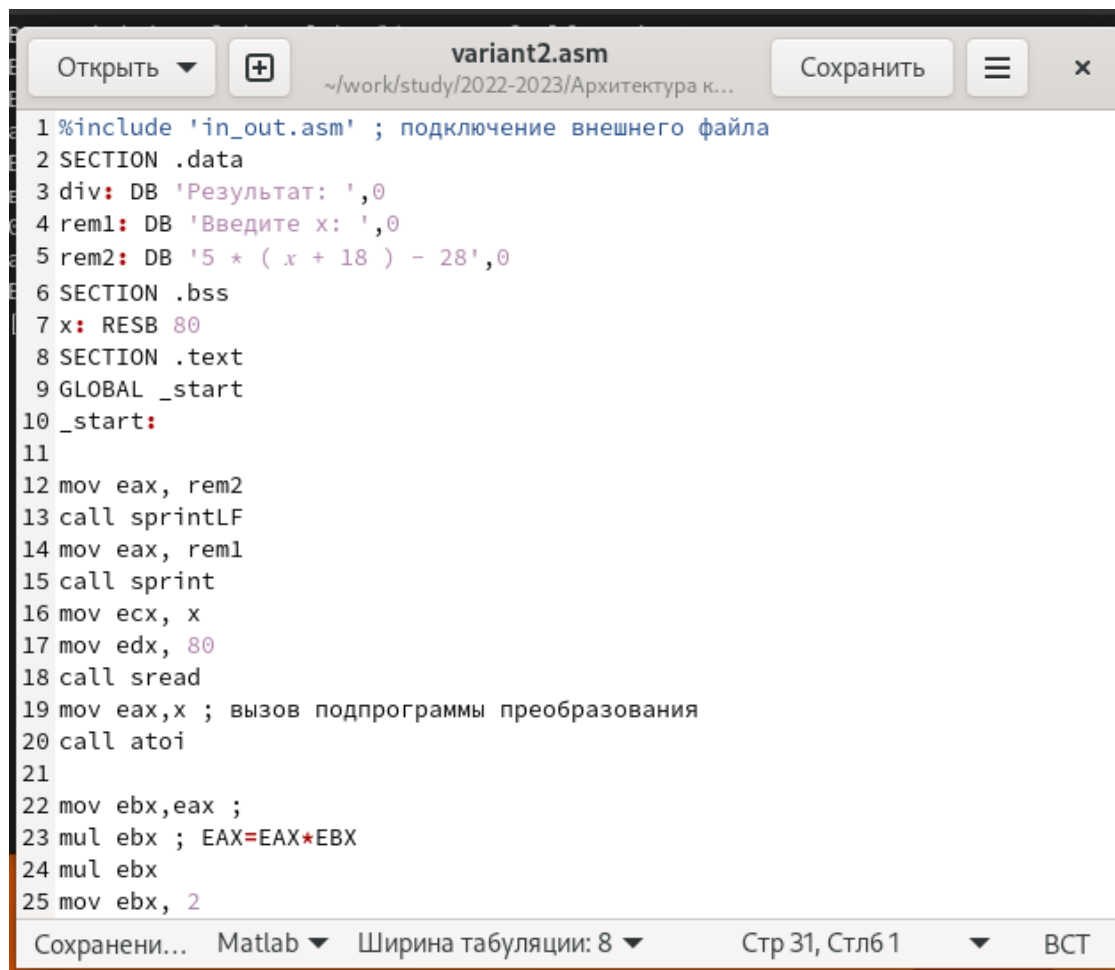
Работа исполняемого файла

Ответы на вопросы лабораторной работы: 1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения 'Ваш вариант'? `mov eax,rem call sprintf` 2. Для чего используется следующие инструкции? `mov ecx, x` - запись входной переменной в

регистр ecx; mov edx, 80 - запись размера переменной в регистр edx; call sread - вызов процедуры чтения данных; 3. Для чего используется инструкция "call atoi"? Вызов atoi – функции преобразующей ascii-код символа в целое число и записывающий результат в регистр eax. 4. Какие строки листинга 7.4 отвечают за вычисления варианта? xor edx,edx mov ebx,20 div ebx inc edx 5. В какой регистр записывается остаток от деления при выполнении инструкции "div ebx"? В регистр ebx. 6. Для чего используется инструкция "inc edx"? Увеличивает значение edx на 1. 7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений? mov eax,rem call sprint mov eax,edx call iprintLF

Задание для самостоятельной работы

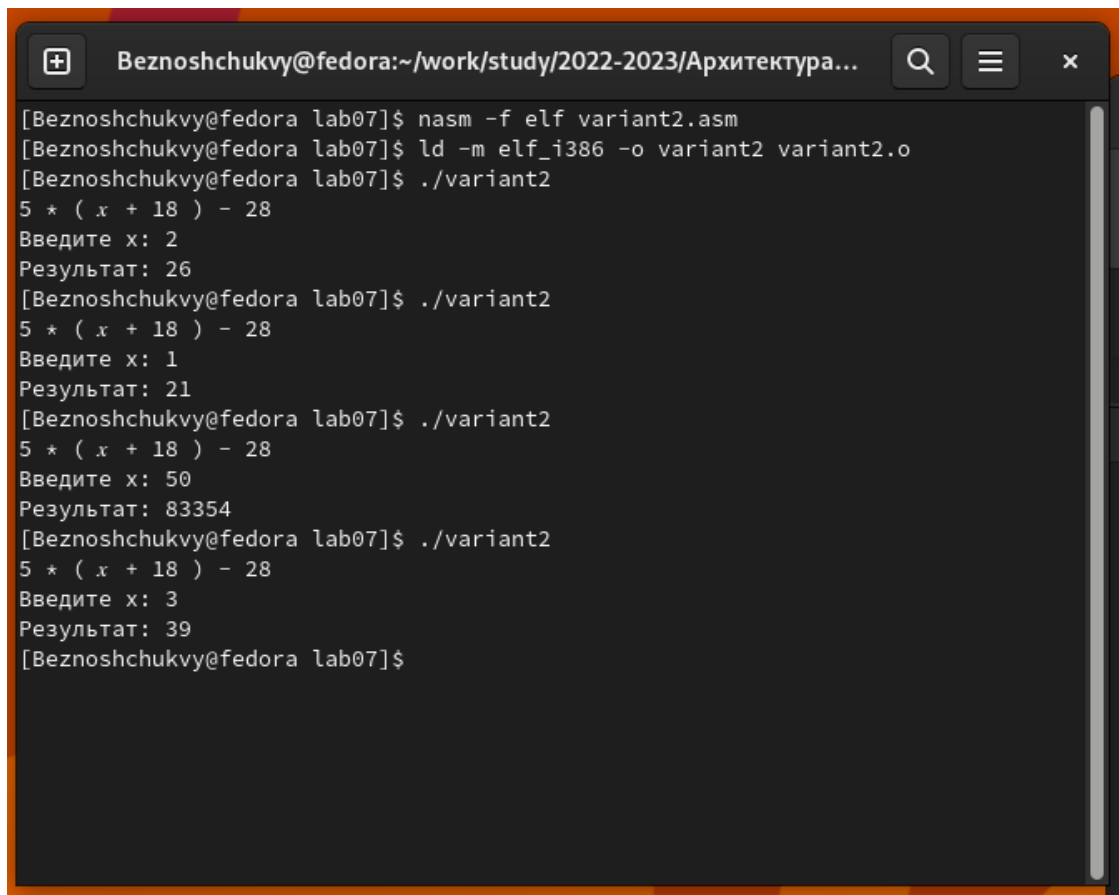
Написали программу вычисления выражения $y = f(x)$. Программа выводит выражение для вычисления, выводит запрос на ввод значения x , вычисляет заданное выражение в зависимости от введенного x , выводит результат вычислений. (рис. [-@fig:021]) Вид функции $f(x)$ выбрали из таблицы 7.3 вариантов заданий, наш номер - 20, полученный при выполнении лабораторной работы. Создали исполняемый файл и проверили его работу для значений $x_1 = 1$ и $x_2 = 3$ из 7.3. (рис. [-@fig:022]).



```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem1: DB 'Введите x: ',0
5 rem2: DB '5 * ( x + 18 ) - 28',0
6 SECTION .bss
7 x: RESB 80
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 mov eax, rem2
13 call sprintf
14 mov eax, rem1
15 call sprintf
16 mov ecx, x
17 mov edx, 80
18 call sread
19 mov eax, x ; вызов подпрограммы преобразования
20 call atoi
21
22 mov ebx, eax ;
23 mul ebx ; EAX=EAX*EBX
24 mul ebx
25 mov ebx, 2
```

Сохранени... Matlab ▾ Ширина табуляции: 8 ▾ Стр 31, Стлб 1 ▾ ВСТ

Файл *variant.asm*

A terminal window with a dark background and orange title bar. The title bar contains a plus icon, the text 'Beznoshchukvy@fedora:~/work/study/2022-2023/Архитектура...', a search icon, a menu icon, and a close icon. The terminal shows the following commands and output:

```
[Beznoshchukvy@fedora lab07]$ nasm -f elf variant2.asm
[Beznoshchukvy@fedora lab07]$ ld -m elf_i386 -o variant2 variant2.o
[Beznoshchukvy@fedora lab07]$ ./variant2
5 * ( x + 18 ) - 28
Введите x: 2
Результат: 26
[Beznoshchukvy@fedora lab07]$ ./variant2
5 * ( x + 18 ) - 28
Введите x: 1
Результат: 21
[Beznoshchukvy@fedora lab07]$ ./variant2
5 * ( x + 18 ) - 28
Введите x: 50
Результат: 83354
[Beznoshchukvy@fedora lab07]$ ./variant2
5 * ( x + 18 ) - 28
Введите x: 3
Результат: 39
[Beznoshchukvy@fedora lab07]$
```

Работа исполняемого файла

Выводы

В результате выполнения лабораторной работы были освоены арифметические инструкции языка ассемблера NASM.