

Шаблон отчёта по лабораторной работе №7

Мугари Абдеррахим , НКАбд-03-22

Содержание

1	Цель работы :	5
2	Выполнение лабораторной работы :	6
2.1	Символьные и численные данные в NASM :	6
2.2	Выполнение арифметических операций в NASM :	14
2.2.1	Вопросы :	20
2.3	Выводы по результатам выполнения заданий :	21
3	Задание для самостоятельной работы :	22
3.1	Выводы по результатам выполнения заданий :	24
4	Выводы, согласованные с целью работы :	25

Список иллюстраций

2.1	Ресунок 1	6
2.2	Ресунок 2	7
2.3	Ресунок 3	7
2.4	Ресунок 4	8
2.5	Ресунок 5	9
2.6	Ресунок 6	10
2.7	Ресунок 7	10
2.8	Ресунок 8	11
2.9	Ресунок 9	12
2.10	Ресунок 10	12
2.11	Ресунок 11	13
2.12	Ресунок 12	14
2.13	Ресунок 13	15
2.14	Ресунок 14	16
2.15	Ресунок 15	17
2.16	Ресунок 16	17
2.17	Ресунок 17	18
2.18	Ресунок 18	18
2.19	Ресунок 19	19
2.20	Ресунок 20	20
3.1	Ресунок 21	22
3.2	Ресунок 22	23
3.3	Ресунок 23	24

Список таблиц

1 Цель работы :

В седьмой лабораторной работе можно будет освоить **арифметические операции** языка ассемблера.

2 Выполнение лабораторной работы :

2.1 Символьные и численные данные в NASM :

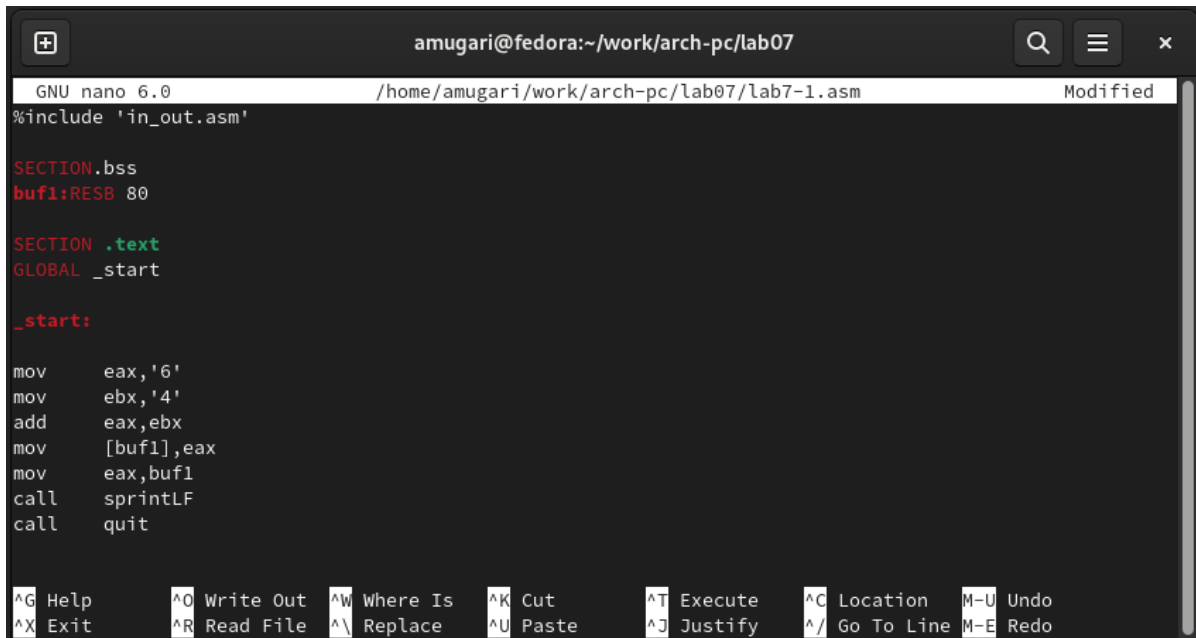
1. Здесь мы начали с создания, а затем переместились в седьмой каталог лаборатории “~/work/arch-pc/lab07”, после чего мы создали файл “lab7-1.asm”. (рис. 2.1)



```
amugari@fedora:~/work/arch-pc/lab07
[amugari@fedora ~]$ mkdir ~/work/arch-pc/lab07
[amugari@fedora ~]$ cd ~/work/arch-pc/lab07
[amugari@fedora lab07]$ touch lab7-1.asm
[amugari@fedora lab07]$
```

Рис. 2.1: Рисунок 1

2. После этого мы заполнили файл **.asm** кодом программы, отображающей значение регистра **eax**. (рис. 2.2)



The screenshot shows a terminal window with the nano text editor open. The title bar indicates the user is 'amugari' on a 'fedora' machine, in the directory '~/work/arch-pc/lab07'. The editor is editing the file '/home/amugari/work/arch-pc/lab07/lab7-1.asm'. The code in the editor is as follows:

```
GNU nano 6.0 /home/amugari/work/arch-pc/lab07/lab7-1.asm Modified
%include 'in_out.asm'

SECTION.bss
buf1:RESB 80

SECTION .text
GLOBAL _start

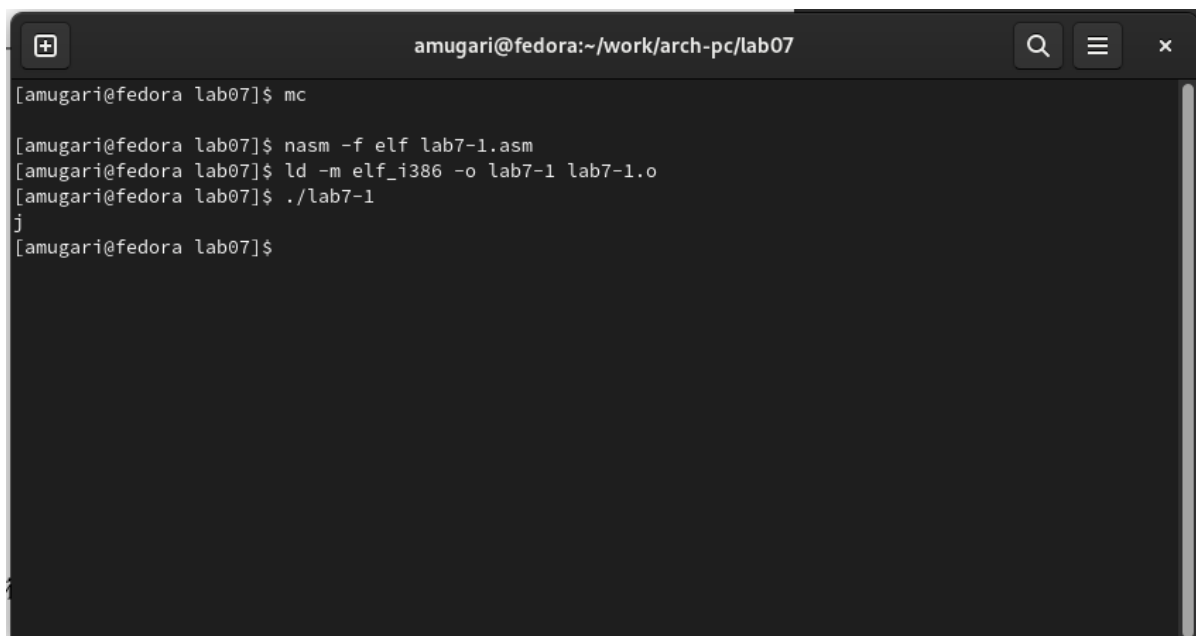
_start:

mov     eax,'6'
mov     ebx,'4'
add     eax,ebx
mov     [buf1],eax
mov     eax,buf1
call    sprintLF
call    quit
```

The bottom status bar of the nano editor shows various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, M-U Undo, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, ^_ Go To Line, and M-E Redo.

Рис. 2.2: Ресунок 2

- Затем мы скомпилировали файл, создали исполняемый файл и запустили программу, все это после перемещения файла **in_out.asm** в тот же каталог, где находится **lab7-1.asm**. (рис. 2.2)



The screenshot shows a terminal window with the user 'amugari' at the 'lab07' prompt. The following commands were entered and executed:

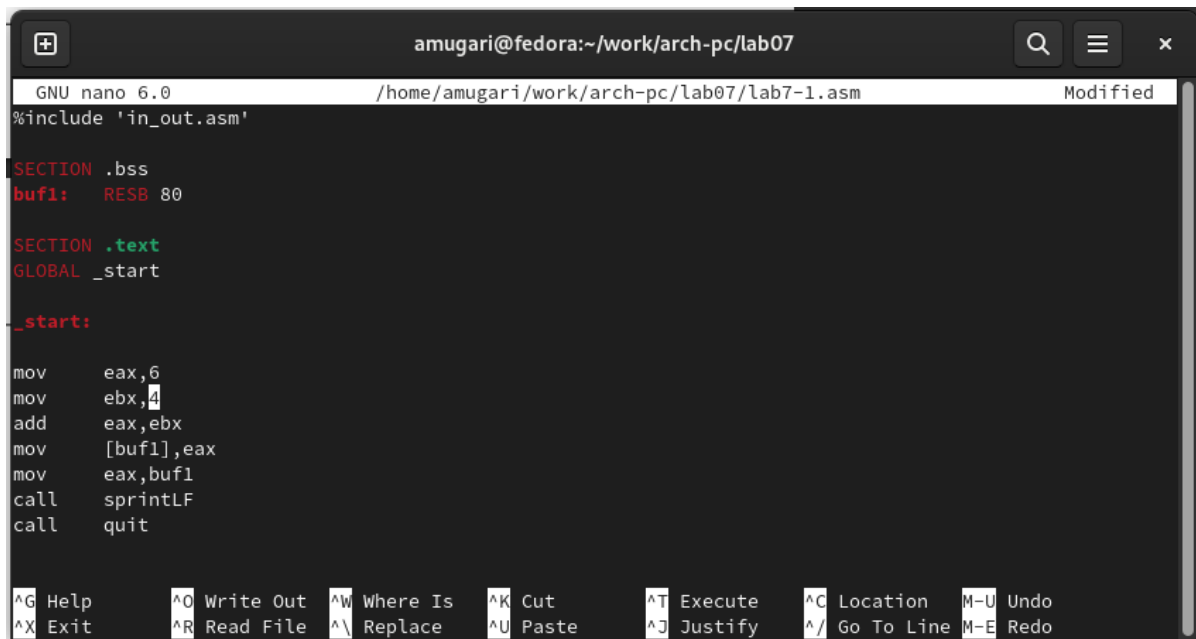
```
[amugari@fedora lab07]$ mc

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

Рис. 2.3: Ресунок 3

3. После этого мы изменили код в листинге следующим образом : `mov eax,6`
`mov ebx,4`

(рис. 2.4)



```
GNU nano 6.0 /home/amugari/work/arch-pc/lab07/lab7-1.asm Modified
#include 'in_out.asm'

SECTION .bss
buf1:  RESB 80

SECTION .text
GLOBAL _start

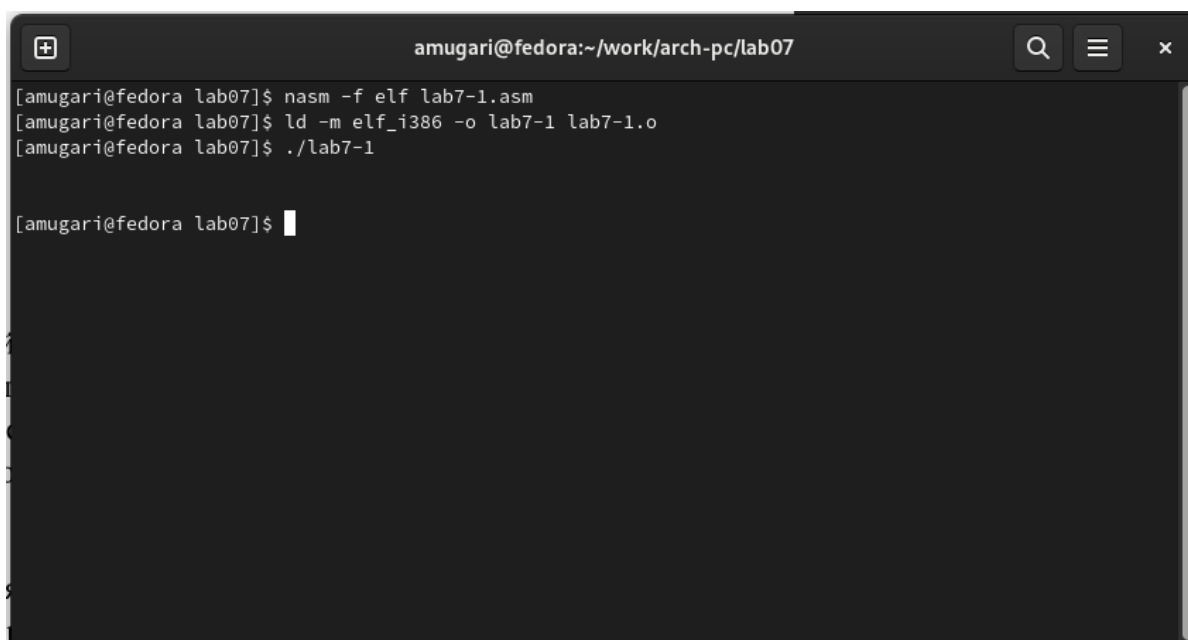
_start:

mov     eax,6
mov     ebx,4
add     eax,ebx
mov     [buf1],eax
mov     eax,buf1
call    printf
call    _exit

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo
```

Рис. 2.4: Ресунок 4

- Затем мы снова скомпилировали файл и создали исполняемый файл. (рис. 2.5)

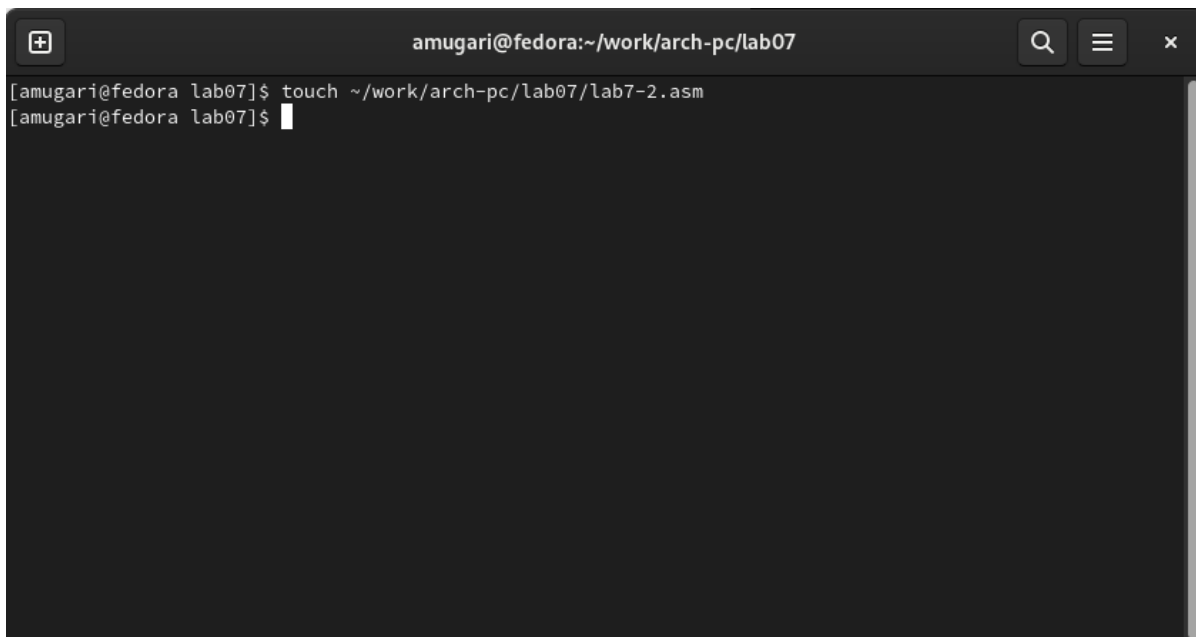
A terminal window titled 'amugari@fedora:~/work/arch-pc/lab07' with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

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

[amugari@fedora lab07]$
```

Рис. 2.5: Ресунок 5

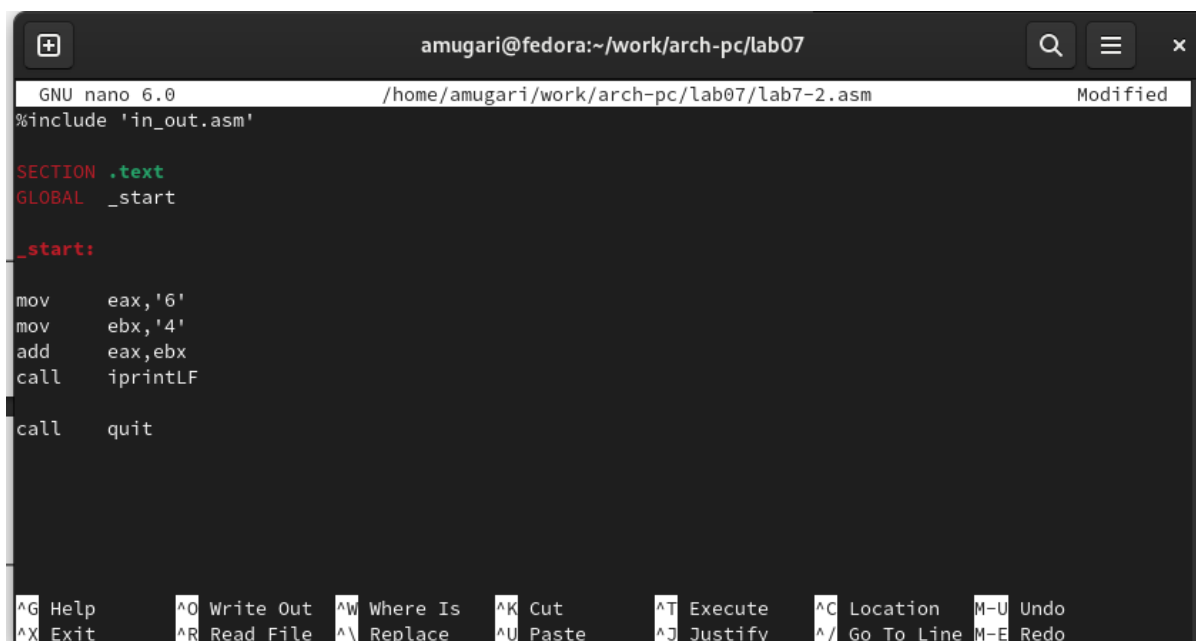
- Проверив **ASCII tbale** символ, соответствующий коду **10** это **новая строка**, и мы можем сказать, что это было отображено, потому что при запуске программы она отобразила новую строку в качестве вывода.
4. После этого мы создали файл **lab-2.asm**, в котором мы использовали под-программы, расположенные в файле **in_out.asm**. (рис. 2.6)

A terminal window with a dark background. The title bar shows 'amugari@fedora:~/work/arch-pc/lab07'. The command prompt is '[amugari@fedora lab07]\$'. The command 'touch ~/work/arch-pc/lab07/lab7-2.asm' has been entered and executed. The prompt is now '[amugari@fedora lab07]\$' with a cursor.

```
amugari@fedora:~/work/arch-pc/lab07
[amugari@fedora lab07]$ touch ~/work/arch-pc/lab07/lab7-2.asm
[amugari@fedora lab07]$
```

Рис. 2.6: Ресунок 6

- После этого мы заполнили файл необходимым кодом для вывода значения реестра с помощью подпрограммы. (рис. 2.7)

A terminal window showing the GNU nano 6.0 text editor. The title bar shows 'amugari@fedora:~/work/arch-pc/lab07'. The editor is editing the file '/home/amugari/work/arch-pc/lab07/lab7-2.asm'. The code in the editor is:

```
GNU nano 6.0 /home/amugari/work/arch-pc/lab07/lab7-2.asm Modified
%include 'in_out.asm'

SECTION .text
GLOBAL _start

_start:

mov     eax,'6'
mov     ebx,'4'
add     eax,ebx
call    iprintLF

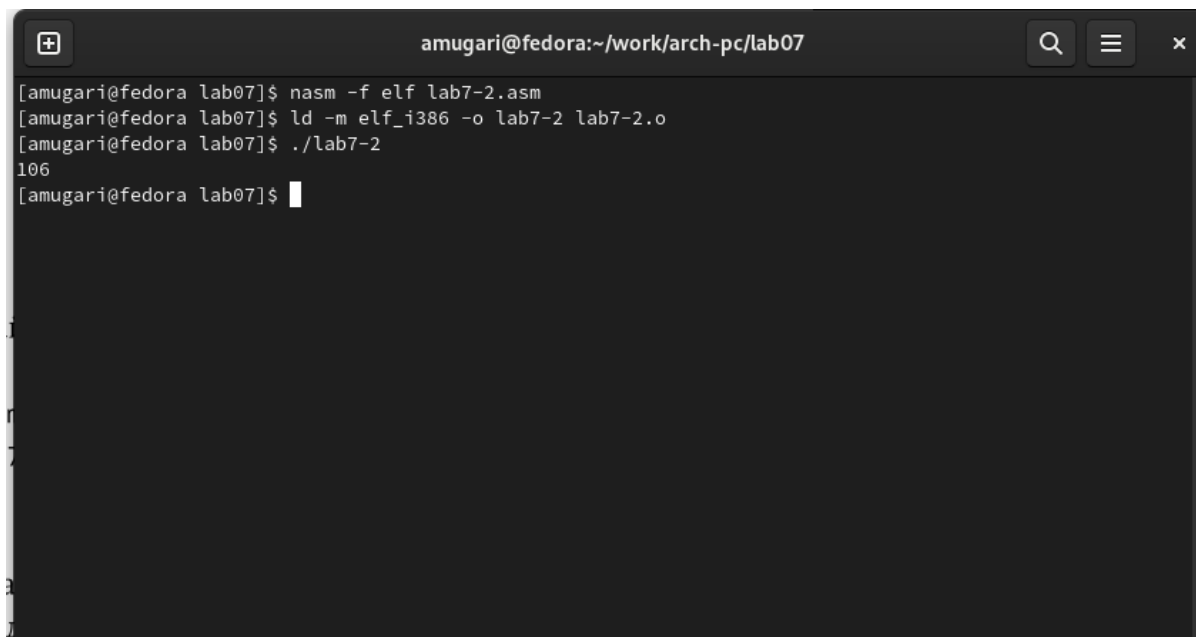
call    quit
```

The bottom status bar shows various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, M-U Undo, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, ^_ Go To Line, M-E Redo.

Рис. 2.7: Ресунок 7

- мы скомпилировали файл, создали исполняемый файл и запустили его.

(рис. 2.8)

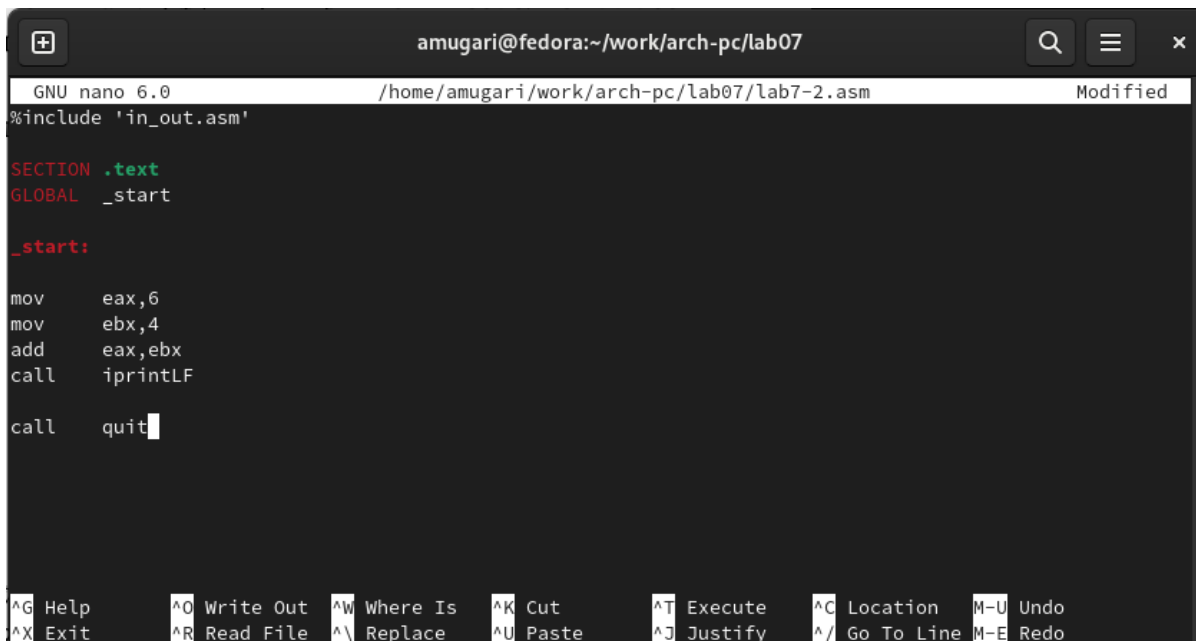
A terminal window with a dark background. The title bar shows the user 'amugari@fedora' and the directory '~/work/arch-pc/lab07'. The terminal contains the following text:

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

Рис. 2.8: Ресунок 8

5. Аналогично предыдущему примеру, мы меняем символы на цифры, заменяя строки на :
- ```
mov eax,6
mov ebx,4
```

. (рис. 2.9)



The screenshot shows a terminal window with the nano text editor open. The title bar indicates the user is 'amugari' on a 'fedora' machine, in the directory '~/work/arch-pc/lab07'. The editor is editing the file '/home/amugari/work/arch-pc/lab07/lab7-2.asm'. The code in the editor is as follows:

```
GNU nano 6.0 /home/amugari/work/arch-pc/lab07/lab7-2.asm Modified
%%include 'in_out.asm'

SECTION .text
GLOBAL _start

_start:

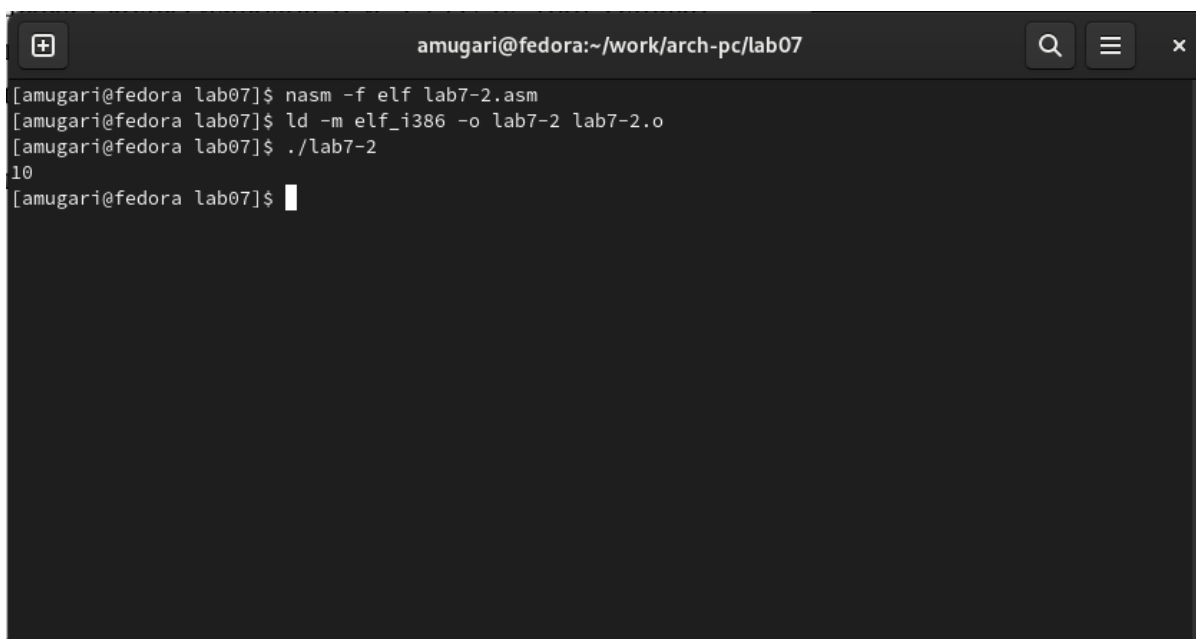
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF

call quit
```

The bottom status bar of the nano editor shows various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, M-U Undo, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, ^\_ Go To Line, M-E Redo.

Рис. 2.9: Ресунок 9

- Затем мы снова скомпилировали файл и создали исполняемый файл. (рис. 2.10)

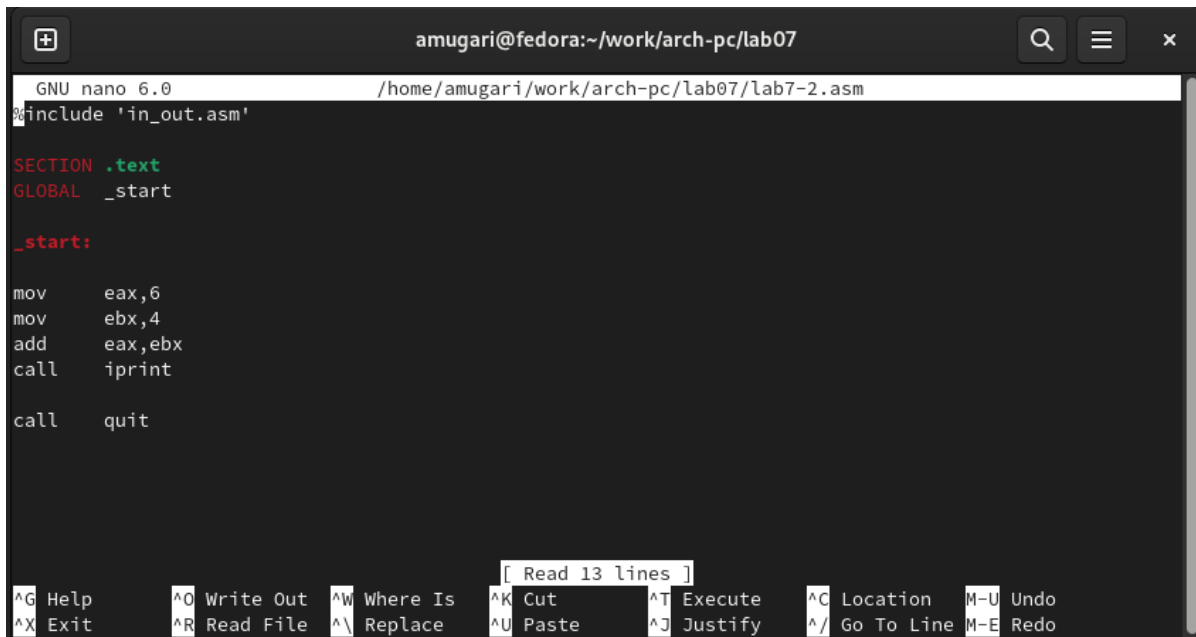


The screenshot shows a terminal window with the user 'amugari' on a 'fedora' machine in the directory '~/work/arch-pc/lab07'. The terminal shows the following commands and output:

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

Рис. 2.10: Ресунок 10

- На этот раз результатом, который мы получили, действительно было добавление 6 и 4 который 10.
- Затем мы заменили функцию **iprintLF** на **iprint**. После этого был создан исполняемый файл, и мы запустили его. (рис. 2.11) (рис. 2.12)



```
GNU nano 6.0 /home/amugari/work/arch-pc/lab07/lab7-2.asm
#include 'in_out.asm'

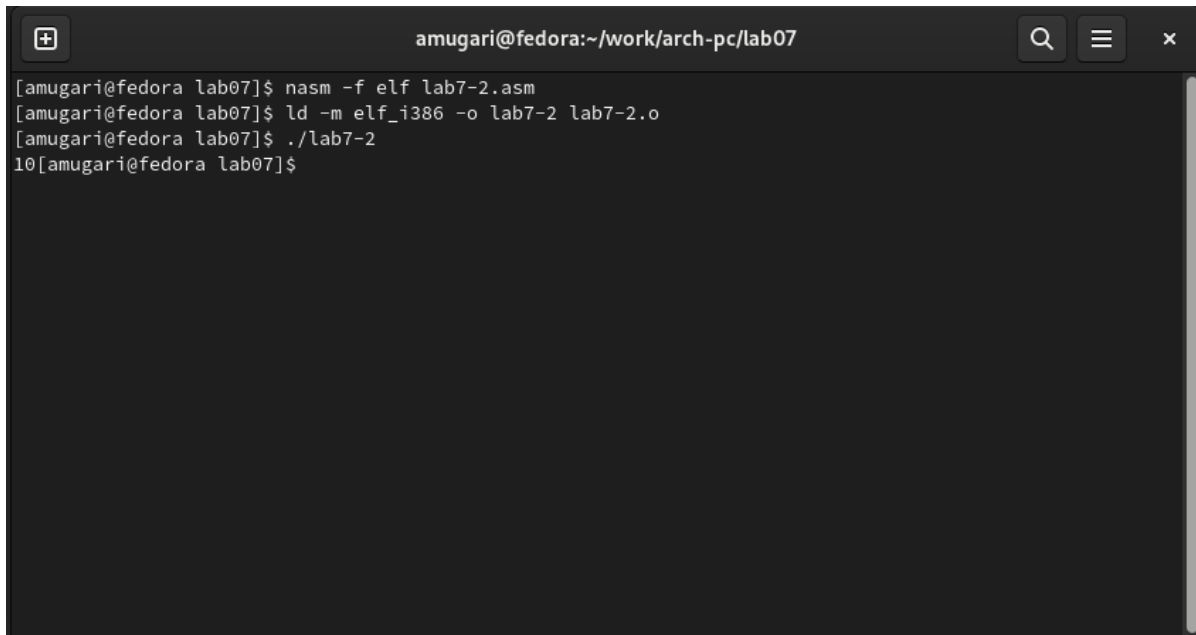
SECTION .text
GLOBAL _start

_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit

[Read 13 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify
^_ ^C Location ^M-U Undo
^/ ^_ Go To Line ^M-E Redo
```

Рис. 2.11: Ресунок 11

A terminal window with a dark background. The title bar shows the user 'amugari' on a 'fedora' machine in the directory '~/work/arch-pc/lab07'. The terminal contains the following text:

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

Рис. 2.12: Ресунок 12

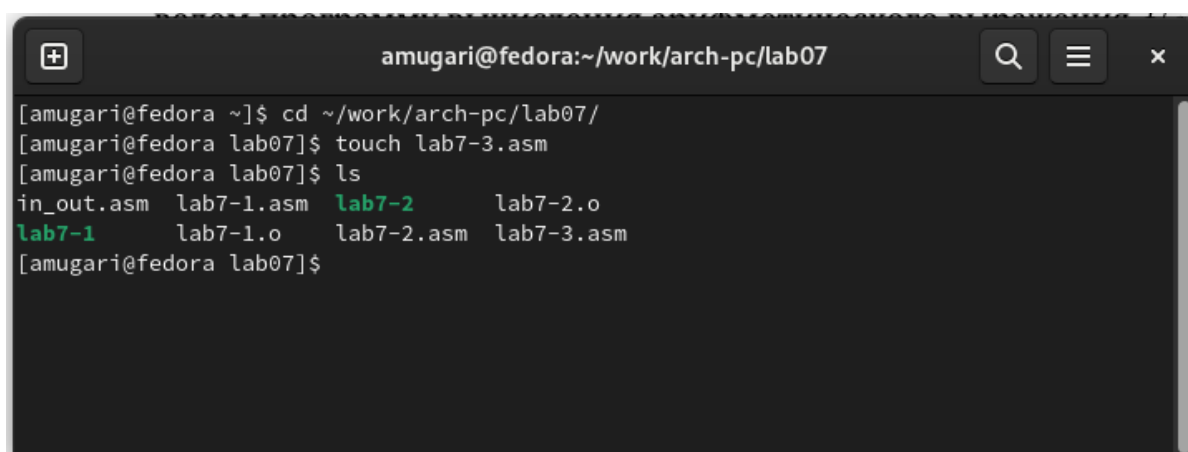
- Когда мы использовали подпрограмму **iprint**, мы заметили, что вывод отличается от предыдущего, потому что при использовании **iprint** не создается новая строка после вывода.

## 2.2 Выполнение арифметических операций в NASM :

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

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

- Мы создали файл lab7-3.asm в каталоге ~/work/arch-pc/lab07. (рис. 2.13)



A terminal window titled "amugari@fedora:~/work/arch-pc/lab07" with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

```
[amugari@fedora ~]$ cd ~/work/arch-pc/lab07/
[amugari@fedora lab07]$ touch lab7-3.asm
[amugari@fedora lab07]$ ls
in_out.asm lab7-1.asm lab7-2 lab7-2.o
lab7-1 lab7-1.o lab7-2.asm lab7-3.asm
[amugari@fedora lab07]$
```

Рис. 2.13: Ресунок 13

2. Затем мы заполнили файл необходимым кодом. (рис. 2.14)

```
mc [amugari@fedora]:~/.work/arch-pc/lab07
GNU nano 6.0 /home/amugari/work/arch-pc/lab07/lab7-3.asm
%include 'in_out.asm'

SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

mov eax,div
call sprintf
mov eax,edi
call iprintLF

mov eax,rem
call sprintf
mov eax,edx
call iprintLF

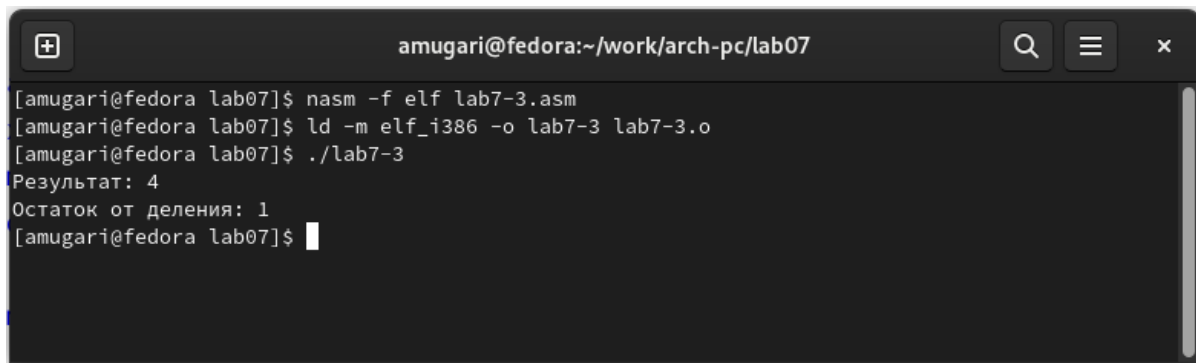
call quit

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^_ Replace ^U Paste ^J Justify ^/ Go To Line
```

Рис. 2.14: Ресунок 14

- Создали исполняемый файл и запустили его. (рис. 2.15)

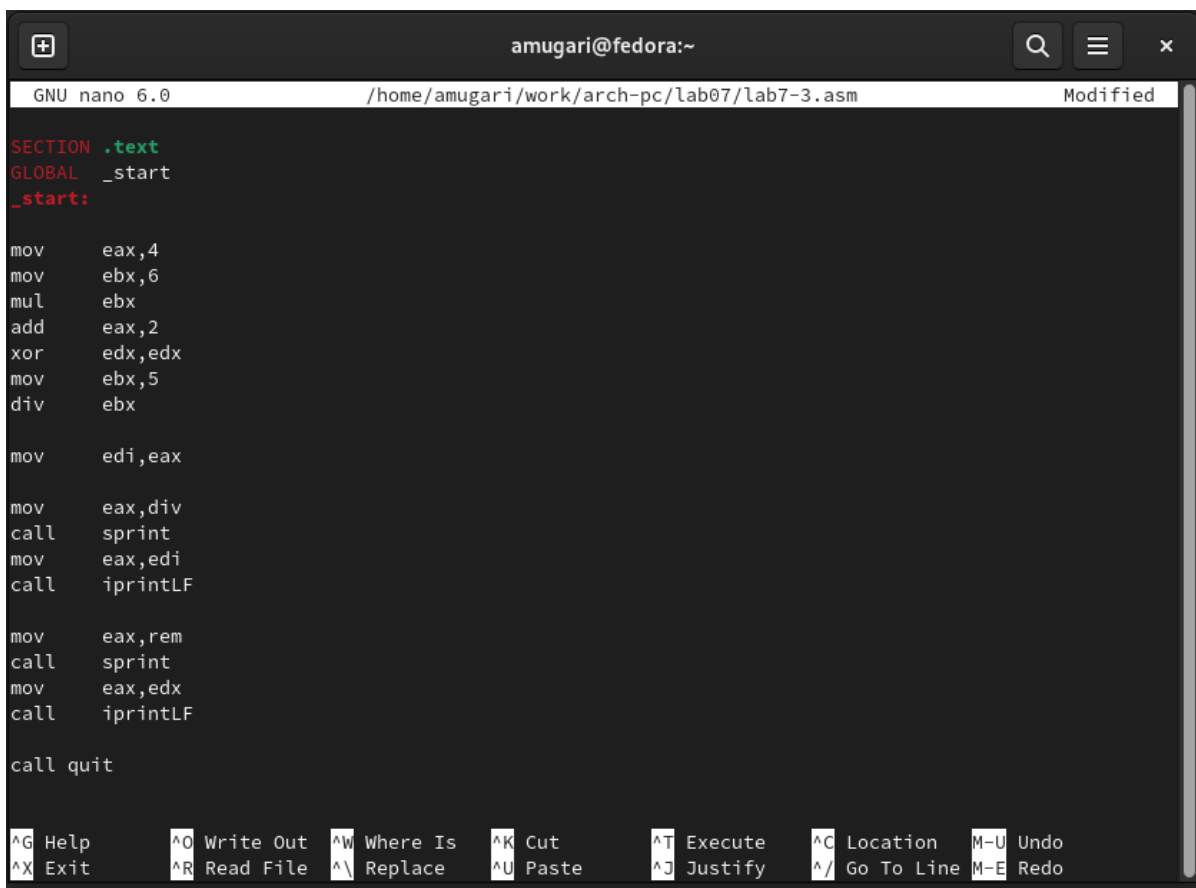




```
amugari@fedora:~/work/arch-pc/lab07
[amugari@fedora lab07]$ nasm -f elf lab7-3.asm
[amugari@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[amugari@fedora lab07]$./lab7-3
Результат: 4
Остаток от деления: 1
[amugari@fedora lab07]$
```

Рис. 2.15: Ресунок 15

- Затем мы изменили текст программы, чтобы вычислить выражение:  
$$f(x) = (4 * 6 + 2) / 5$$
  
(рис. 2.16)



```
GNU nano 6.0 /home/amugari/work/arch-pc/lab07/lab7-3.asm Modified
SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

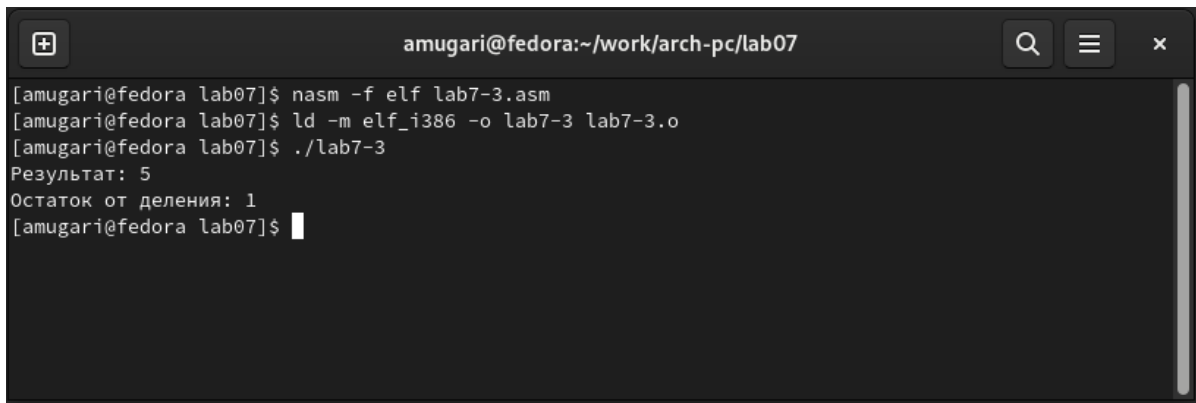
mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo
```

Рис. 2.16: Ресунок 16

- мы создали исполняемый файл и проверили его работу. (рис. 2.17)

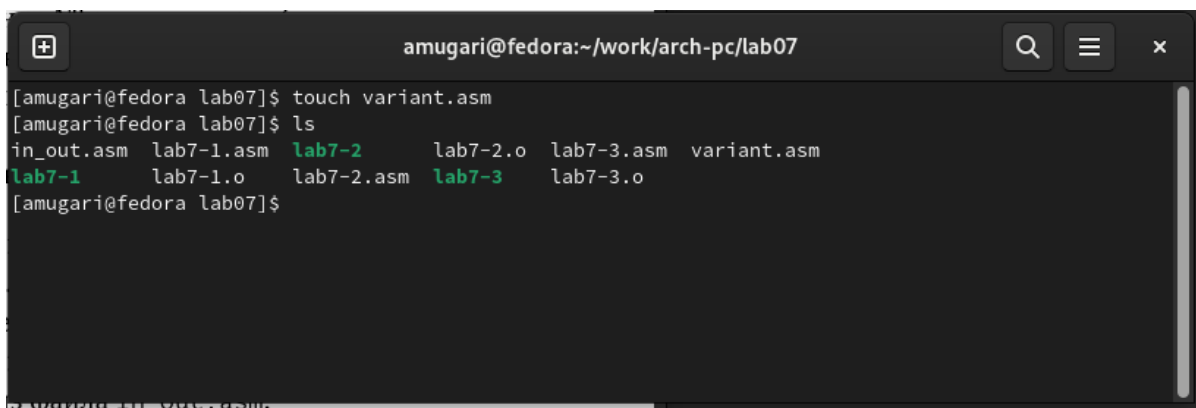


```
amugari@fedora:~/work/arch-pc/lab07
[amugari@fedora lab07]$ nasm -f elf lab7-3.asm
[amugari@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[amugari@fedora lab07]$./lab7-3
Результат: 5
Остаток от деления: 1
[amugari@fedora lab07]$
```

Рис. 2.17: Ресунок 17

3. На этом шаге мы написали программу, которая может вычислить дисперсию, которую мы получаем из номера студенческого билета.

- Мы начали с создания файла **variant.asm**. (рис. 2.18)



```
amugari@fedora:~/work/arch-pc/lab07
[amugari@fedora lab07]$ touch variant.asm
[amugari@fedora lab07]$ ls
in_out.asm lab7-1.asm lab7-2 lab7-2.o lab7-3.asm variant.asm
lab7-1 lab7-1.o lab7-2.asm lab7-3 lab7-3.o
```

Рис. 2.18: Ресунок 18

- После этого мы написали код программы. (рис. 2.19)

```
GNU nano 6.0 /home/amugari/work/arch-pc/lab07/variant.asm Modified
%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 sprintf

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

xor edx, edx
mov ebx, 20
div ebx
inc edx

mov eax, rem
call sprintf
mov eax, edx
call iprintLF

call quit

File Name to Write: /home/amugari/work/arch-pc/lab07/variant.asm
^G Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend ^T Browse
```

Рис. 2.19: Ресунок 19

- мы создали исполняемый файл и проверили его работу, и действительно, в зависимости от номера студента он генерирует номер варианта. (рис. 2.20)

A screenshot of a terminal window with a dark background. The title bar shows the user 'amugari' on a 'fedora' machine in the directory '~/work/arch-pc/lab07'. The terminal contains the following text:

```
[amugari@fedora lab07]$ nasm -f elf variant.asm
[amugari@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[amugari@fedora lab07]$./variant
Введите № студенческого билета:
1032215692
Ваш вариант: 13
[amugari@fedora lab07]$
```

Рис. 2.20: Ресунок 20

### 2.2.1 Вопросы :

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

**О:**

```
rem: DB ‘Ваш вариант:’,0
mov eax,rem
call sprint
```

2. Для чего используются следующие инструкции?

```
mov ecx, x
mov edx, 80
call sread
```

**О:** Эти инструкции были использованы для того, чтобы позволить пользователю вводить данные

3. Для чего используется инструкция “call atoi”?

**О:** Эта инструкция используется для преобразования значения x из **ASCII-кода** в целое число.

4. Какие строки листинга 7.4 отвечают за вычисления варианта?

**О:**

```
xor edx,edx
mov ebx,20
div ebx
inc edx
```

5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

**О:** Остаток был записан в регистре **edx**

6. Для чего используется инструкция “inc edx”?

**О:** Эта инструкция была использована для увеличения значения в регистре **edx**

7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений?

**О:**

```
mov eax,edx
call iprintLF
```

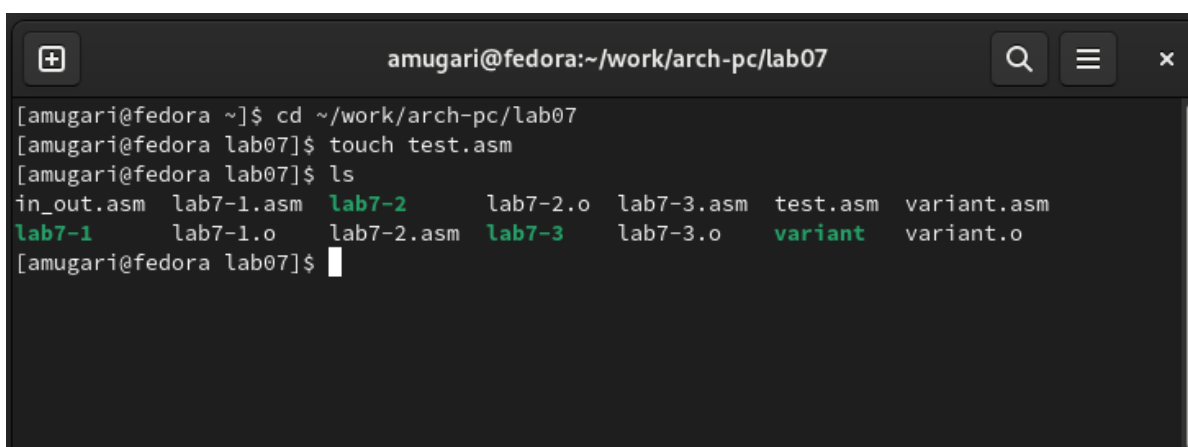
## 2.3 Выводы по результатам выполнения заданий :

- В ходе лабораторной работы мы освоили выполнение арифметических операций на языке ассемблера и углубились в использование подпрограммы.

### 3 Задание для самостоятельной работы :

1. В этой работе нам пришлось написать программу, которая просит пользователя ввести значение переменной и решить математическое выражение.

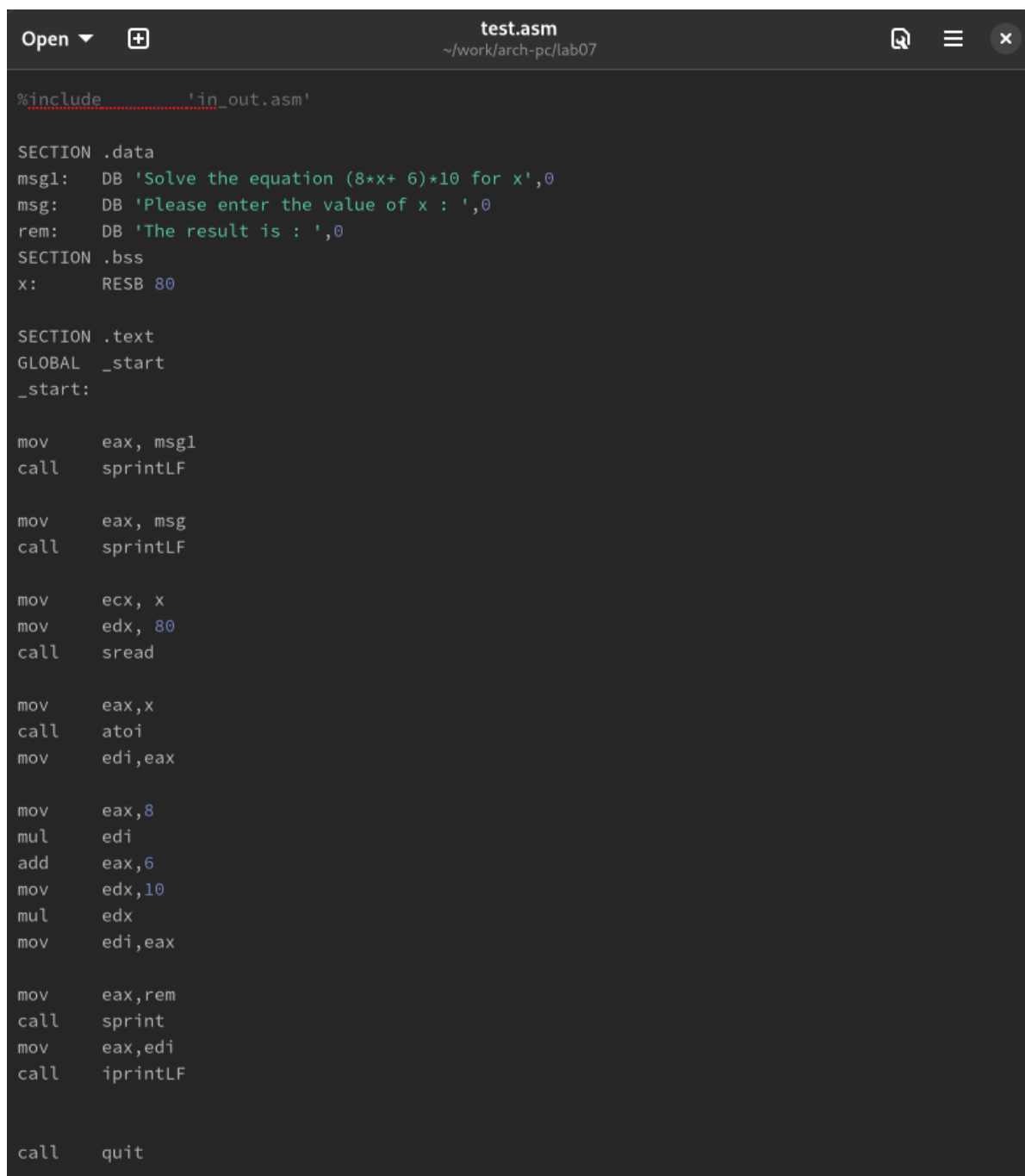
- Мой вариант : 13
- математическое выражение  $(8x + 6) \cdot 10$
- Итак, мы начали с создания asm-файла, в котором будет находиться наш код. (рис. 3.1)



```
amugari@fedora:~/work/arch-pc/lab07
[amugari@fedora ~]$ cd ~/work/arch-pc/lab07
[amugari@fedora lab07]$ touch test.asm
[amugari@fedora lab07]$ ls
in_out.asm lab7-1.asm lab7-2 lab7-2.o lab7-3.asm test.asm variant.asm
lab7-1 lab7-1.o lab7-2.asm lab7-3 lab7-3.o variant variant.o
[amugari@fedora lab07]$
```

Рис. 3.1: Ресунок 21

- После этого мы написали код нашей программы. (рис. 3.2)



```
Open ▾ + test.asm
~/work/arch-pc/lab07

%include 'in_out.asm'

SECTION .data
msg1: DB 'Solve the equation (8*x+ 6)*10 for x',0
msg: DB 'Please enter the value of x : ',0
rem: DB 'The result is : ',0
SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg1
call sprintf

mov eax, msg
call sprintf

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi
mov edi, eax

mov eax, 8
mul edi
add eax, 6
mov edx, 10
mul edx
mov edi, eax

mov eax, rem
call sprintf
mov eax, edi
call iprintLF

call quit
```

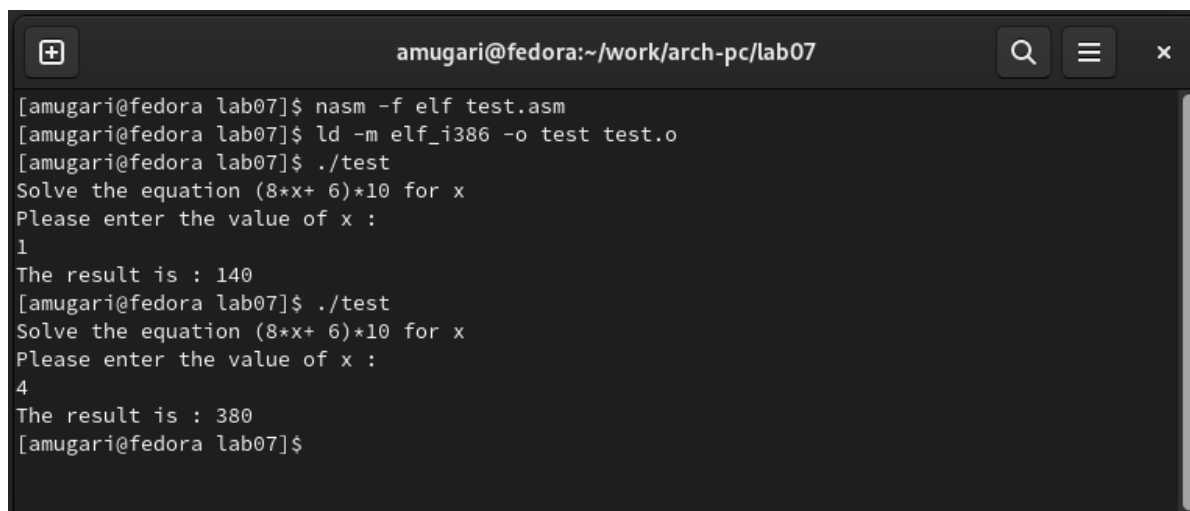
Рис. 3.2: Рисунок 22

- и, наконец, мы проверяем корректность кода, который мы написали, используя два разных значения

$$x_1 = 1$$

$$x_2 = 4$$

Как указано на следующем рисунке (рис. 3.3)

A screenshot of a terminal window with a dark background. The window title is 'amugari@fedora:~/work/arch-pc/lab07'. The terminal shows the following sequence of commands and output:

```
[amugari@fedora lab07]$ nasm -f elf test.asm
[amugari@fedora lab07]$ ld -m elf_i386 -o test test.o
[amugari@fedora lab07]$./test
Solve the equation (8*x+ 6)*10 for x
Please enter the value of x :
1
The result is : 140
[amugari@fedora lab07]$./test
Solve the equation (8*x+ 6)*10 for x
Please enter the value of x :
4
The result is : 380
[amugari@fedora lab07]$
```

Рис. 3.3: Ресунок 23

### 3.1 Выводы по результатам выполнения заданий :

В этой части мы смогли узнать, как преобразовать некоторые математические идеи в реальный код на ассемблере, что помогло нам получить более глубокое представление о том, как работать с регистрами.



## **4 Выводы, согласованные с целью работы :**

- В седьмой лаборатории мы в основном научились писать программы, выполняющие арифметические операции, и научились вычислять математические выражения средней сложности.