

Отчет по лабораторной работе No.6

Дисциплины: Архитектура компьютера

Нджову Нелиа

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	19
	Список литературы	20

Список иллюстраций

3.1	Рис 1	7
3.2	Рис 2	7
3.3	Рис 3	8
3.4	Рис 4	8
3.5	Рис 5	9
3.6	Рис 6	9
3.7	Рис 7	9
3.8	Рис 8	10
3.9	Рис 9	10
3.10	Рис 10	11
3.11	Рис 11	11
3.12	Рис 12	11
3.13	Рис 13	12
3.14	Рис 14	12
3.15	Рис 15	13
3.16	Рис 16	13
3.17	Рис 17	13
3.18	Рис 18	14
3.19	Рис 19	14
3.20	Рис 20	15
3.21	Рис 21	16
3.22	Рис 22	16
3.23	Рис 23	17

Список таблиц

1 Цель работы

Целью данной работы является освоение арифметических инструкций на языке ассемблера NASM.

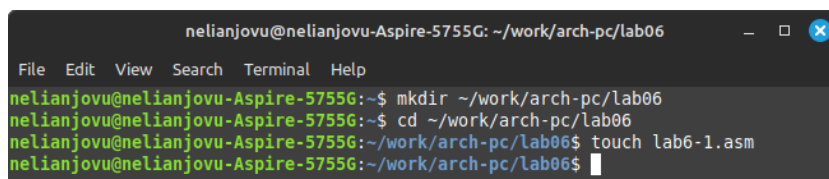
2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

3 Выполнение лабораторной работы

1. Символьные и численные данные в NASM

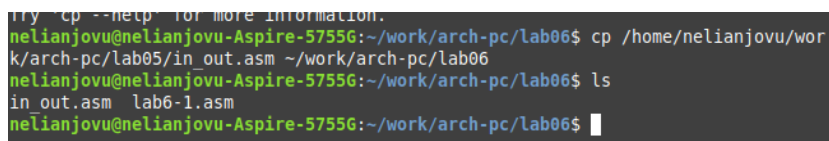
Я создам каталог для программ лабораторных работ 6 с помощью команды `mkdir`, зайду в него с помощью команды `cd` и создам файл `lab6-1.asm` с помощью команды `touch`(рис 1)



```
nelianjovu@nelianjovu-Aspire-5755G: ~/work/arch-pc/lab06
File Edit View Search Terminal Help
nelianjovu@nelianjovu-Aspire-5755G:~$ mkdir ~/work/arch-pc/lab06
nelianjovu@nelianjovu-Aspire-5755G:~$ cd ~/work/arch-pc/lab06
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ touch lab6-1.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$
```

Рис. 3.1: Рис 1

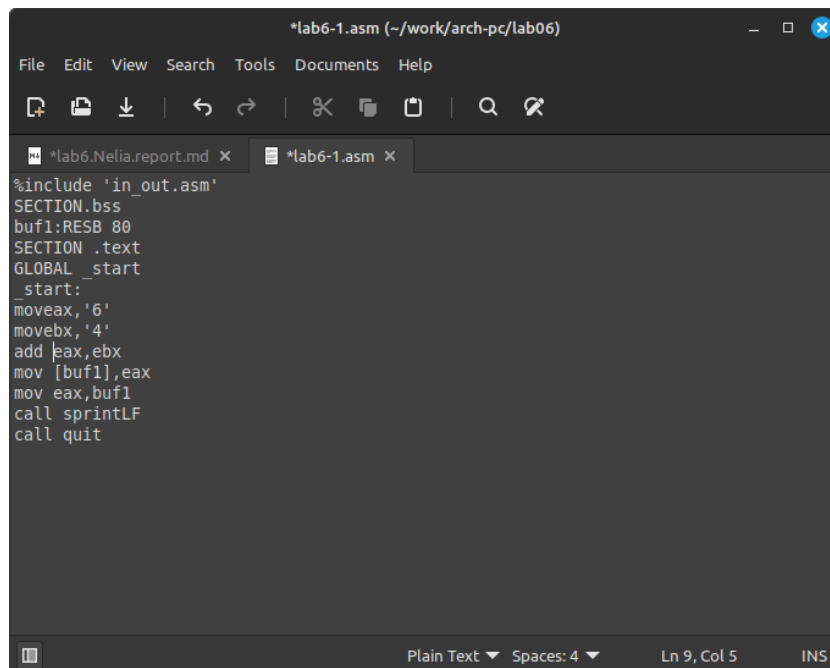
Я скопирую файл `in_out.asm` в текущий каталог с помощью команды `cp`, потому что буду использовать его в программах(рис 2)



```
Try 'cp --help' for more information.
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ cp /home/nelianjovu/work/arch-pc/lab05/in_out.asm ~/work/arch-pc/lab06
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$
```

Рис. 3.2: Рис 2

Открою созданный файл `lab6-1.asm` и скопирую в него программу вывода значения регистра `eax`(рис 3)



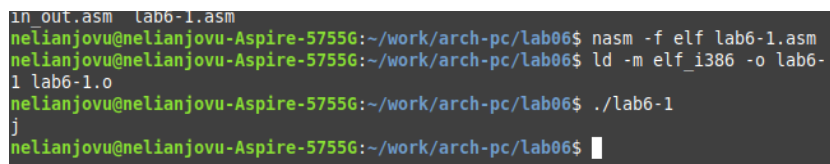
The screenshot shows a text editor window titled '*lab6-1.asm (~/.work/arch-pc/lab06)'. The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for file operations and editing. The editor has two tabs: '*lab6.Nelia.report.md' and '*lab6-1.asm'. The code in the active tab is as follows:

```
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
    mov eax, '6'
    mov ebx, '4'
    add ebx, ebx
    mov [buf1], ebx
    mov eax, buf1
    call sprintf
    call quit
```

The status bar at the bottom indicates 'Plain Text', 'Spaces: 4', 'Ln 9, Col 5', and 'INS'.

Рис. 3.3: Рис 3

Я создам исполняемый файл программы и запущу его. Программа выведет символ j, поскольку программа выводит символ, соответствующий сумме ASCII двоичных кодов символов 4 и 6(рис 4)

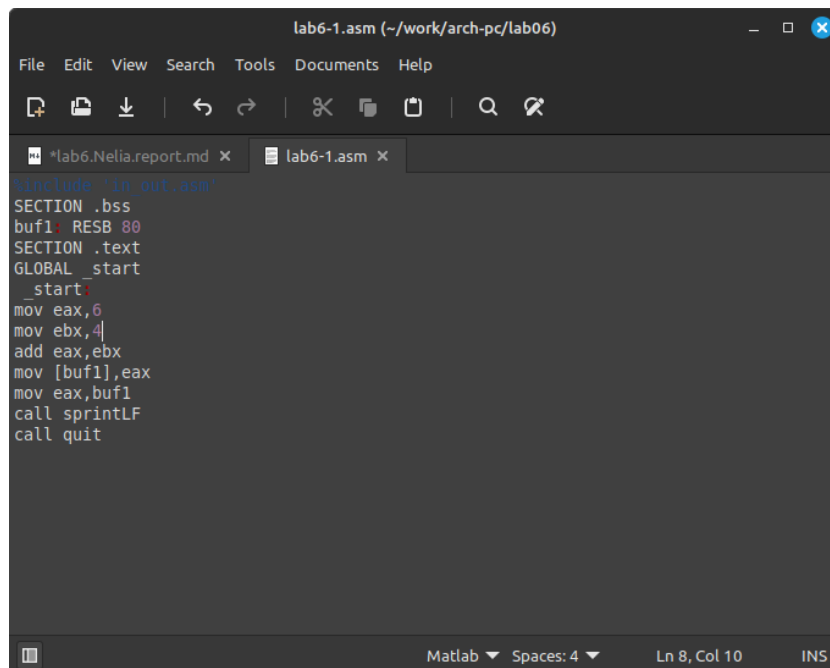


The screenshot shows a terminal window with the following commands and output:

```
in_out.asm lab6-1.asm
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$ ./lab6-1
j
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$
```

Рис. 3.4: Рис 4

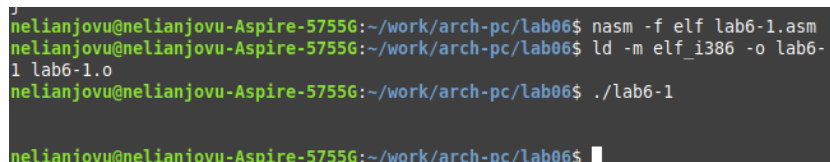
Я заменю символы '6' и '4' в тексте программы на цифры 6 и 4(рис 5)



```
lab6-1.asm (~/.work/arch-pc/lab06)
File Edit View Search Tools Documents Help
*lab6.Nelia.report.md x lab6-1.asm x
%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 sprintf
call quit
Matlab Spaces: 4 Ln 8, Col 10 INS
```

Рис. 3.5: Рис 5

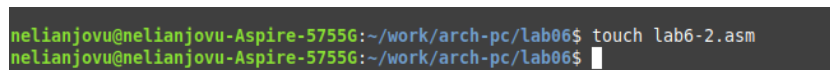
Я создам новый исполняемый файл программы и запущу его. Теперь отображается символ с кодом 10, это символ перевода строки(рис 6)



```
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$ ./lab6-1
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$
```

Рис. 3.6: Рис 6

Я создам новый файл lab6-2.asm в каталоге ~/.work/arch-pc/lab06 с помощью команды touch(рис 7)

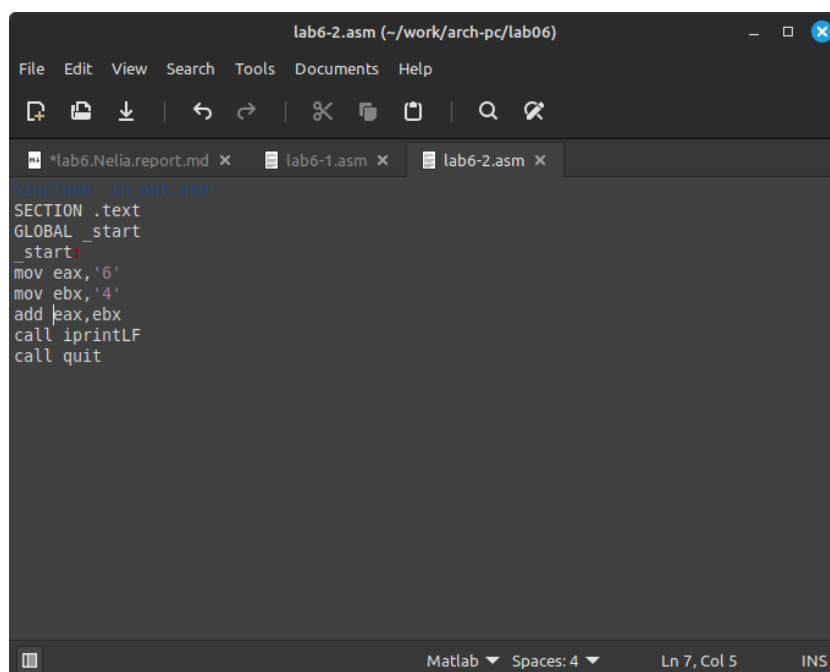


```
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$ touch lab6-2.asm
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$
```

Рис. 3.7: Рис 7

Я открою вновь созданный файл и скопирую в него заданный текст програм-

мы(рис 8)



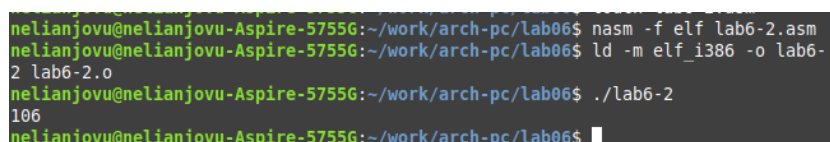
The screenshot shows a text editor window titled 'lab6-2.asm (~/.work/arch-pc/lab06)'. The editor has a menu bar with 'File', 'Edit', 'View', 'Search', 'Tools', 'Documents', and 'Help'. Below the menu is a toolbar with icons for file operations and editing. The main text area contains the following assembly code:

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

The status bar at the bottom indicates 'Matlab', 'Spaces: 4', 'Ln 7, Col 5', and 'INS'.

Рис. 3.8: Рис 8

Я создаю и запускаю исполняемый файл lab6-2. Теперь на выходе будет число 106, поскольку функция iprintLF позволяет программе выводить точное число вместо символа ASCII(рис 9)



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

```
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ./lab6-2
106
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$
```

Рис. 3.9: Рис 9

Теперь я заменяю символы '6' и '4' в тексте программы на цифры 6 и 4(рис 10)

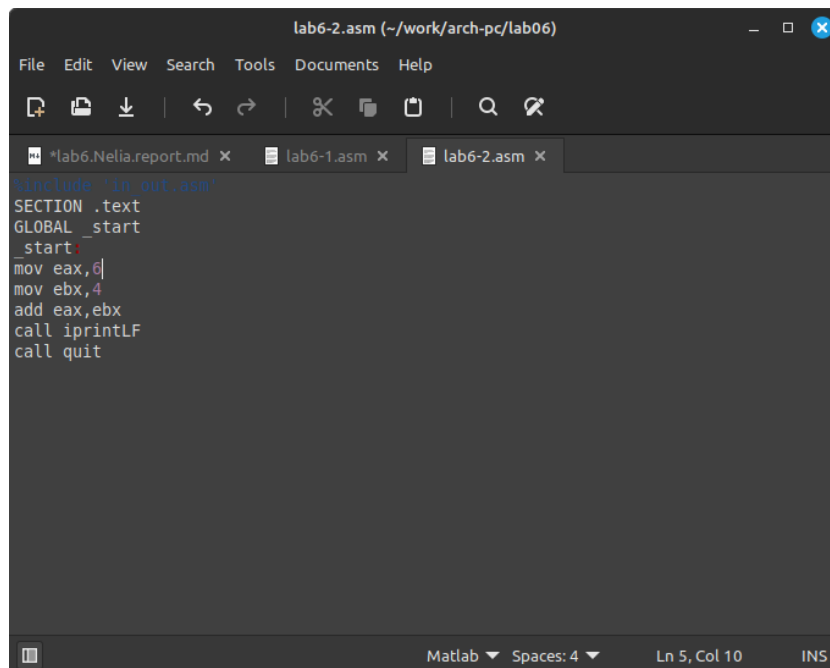


Рис. 3.10: Рис 10

Как и ранее, я создаю и запускаю исполняемый файл lab6-2. Вывод равен 10, поскольку функция `iprintLF` позволяет программе выводить точное число вместо символа ASCII(рис 11)

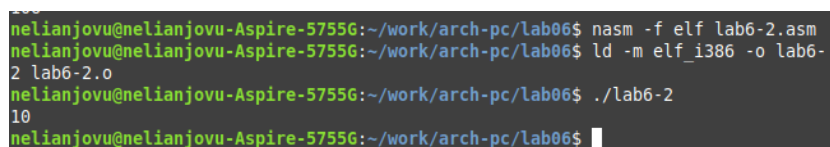


Рис. 3.11: Рис 11

2. Выполнение арифметических операций в NASM

Я создам еще один файл lab6-3.asm в каталоге `~/work/arch-pc/lab06` с помощью команды `touch`(рис 12)

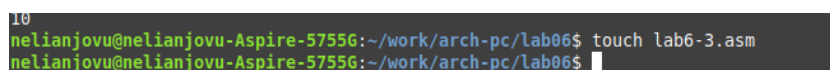
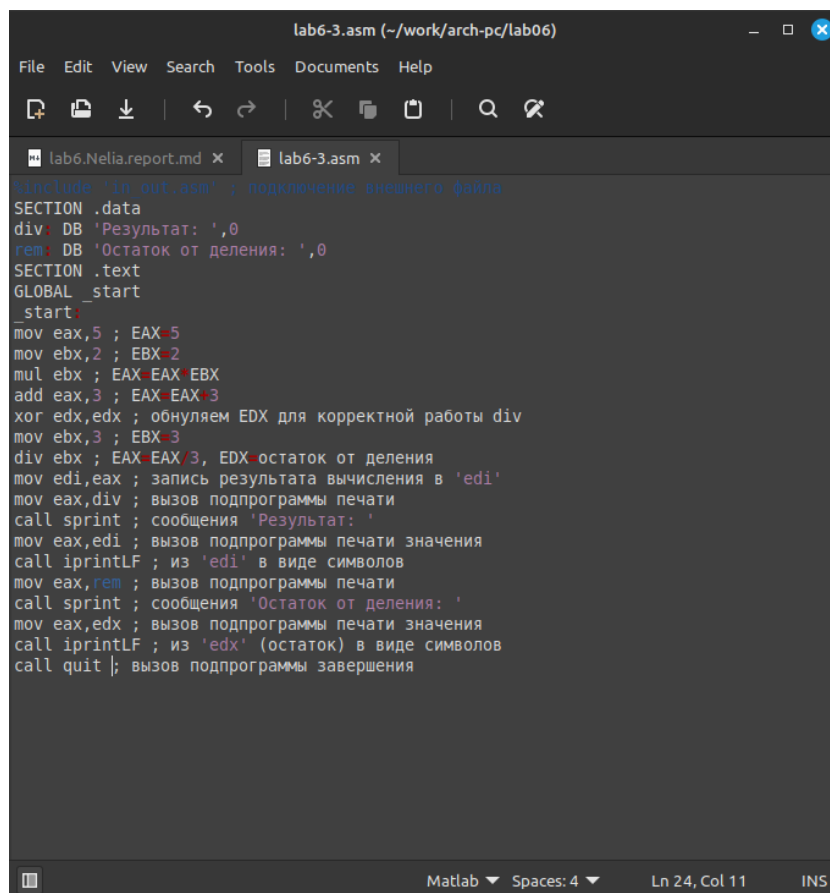


Рис. 3.12: Рис 12

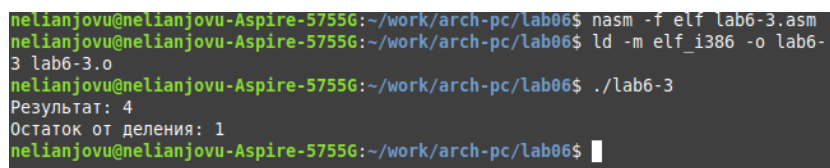
В созданный файл ввожу текст программы для вычисления значения выражения $f(x) = (5 * 2 + 3)/3$ (рис 13)



```
lab6-3.asm (~/.work/arch-pc/lab06)
File Edit View Search Tools Documents Help
lab6.Nelia.report.md x lab6-3.asm x
#include "in_out.asm" ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
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'
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintfLF ; 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 iprintfLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 3.13: Рис 13

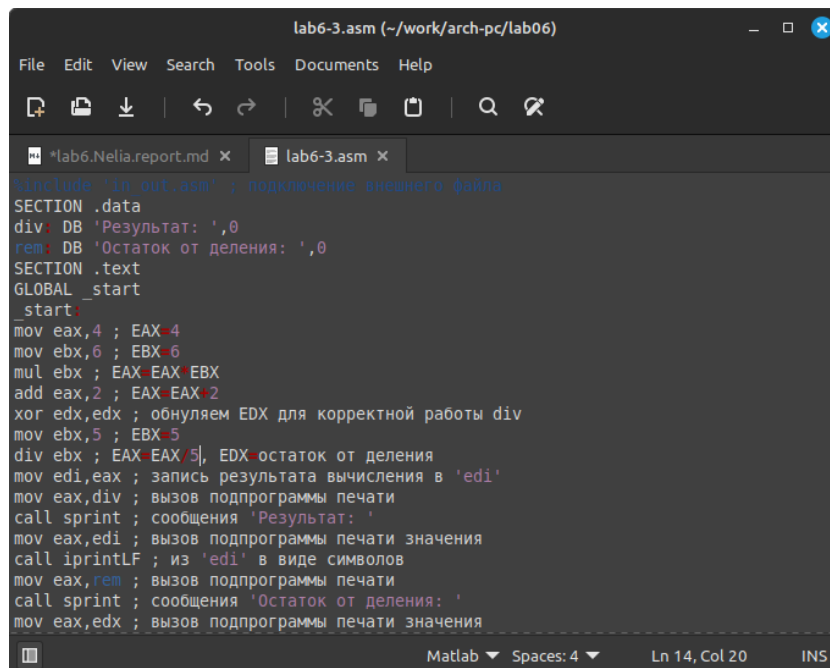
Создаю исполняемый файл и запускаю его (рис 14)



```
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$
```

Рис. 3.14: Рис 14

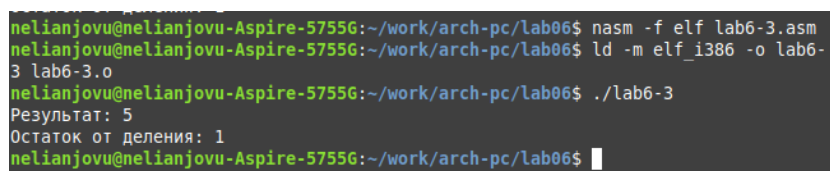
Я изменю программу так, чтобы она вычисляла значение выражения $f(x) = (4 * 6 + 2)/5$ (рис 15)



```
lab6-3.asm (~/work/arch-pc/lab06)
File Edit View Search Tools Documents Help
*lab6.Nelia.report.md x lab6-3.asm x
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
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'
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
```

Рис. 3.15: Рис 15

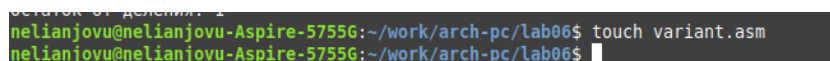
Теперь я создаю исполняемый файл и запускаю его(рис 16)



```
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$
```

Рис. 3.16: Рис 16

Я создам файл variant.asm в каталоге ~/work/arch-pc/lab06 с помощью команду touch(рис 17)



```
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ touch variant.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$
```

Рис. 3.17: Рис 17

Скопирую текст программы в файл для расчета варианта задания по студенческому билету(рис 18)

```

variant.asm (-/work/arch-pc/lab06)
File Edit View Search Tools Documents Help
*lab6.Nelia.report.md x lab6-3.asm x variant.asm x
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите No 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 ; 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 sprintf
mov ebx, edx
call iprintf
call quit

```

Рис. 3.18: Рис 18

Я создам и запущу исполняемый файл. Ввожу с клавиатуры номер студенческого билета, программа показывает, что мой вариант-14(рис 19)

```

nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ nasm -f elf variant.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant.o
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ./variant
Введите No студенческого билета:
1032239033
Ваш вариант: 14
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$

```

Рис. 3.19: Рис 19

2.1 Ответы на вопросы по программе

1.mov eax,rem
call sprint

2. `mov ecx,x` — используется для помещения адреса входной строки `x` в регистр `ecx`.

`mov edx,80` — записывает длину входной строки в регистр `edx`

`call sread` — вызов подпрограммы из внешнего файла, позволяющей ввести сообщение с клавиатуры

3. `call atoi` используется для преобразования `ascii`-кода символа в целое число и записи результата в регистр `eax`

`4.xor edx,edx`

`mov ebx,20`

`div ebx`

`inc edx`

5. Остаток от деления записывается в регистр `edx`

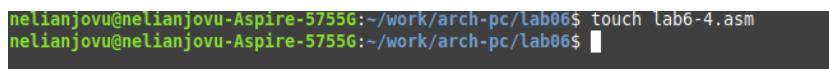
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1

7. `mov eax,edx`

`call iprintLF`

3. Выполнение заданий для самостоятельной работы

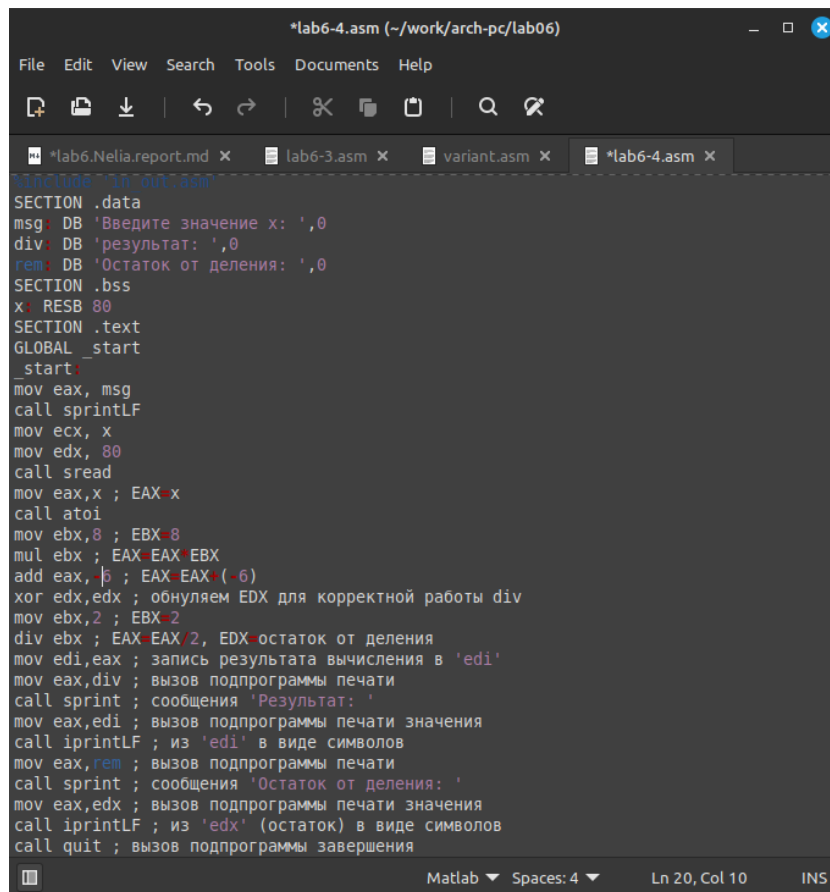
Я создам файл `lab6-4.asm` с помощью команды `touch` (рис 20)



```
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$ touch lab6-4.asm
nelianjovu@nelianjovu-Aspire-57556:~/work/arch-pc/lab06$
```

Рис. 3.20: Рис 20

Созданный файл открою для редактирования, введу в него текст программы для вычисления значения выражения $(8 \cdot x - 6) / 2$ (рис. 4.24). Это выражение было под вариантом 12 (рис 21)



```
*lab6-4.asm (~/.work/arch-pc/lab06)
File Edit View Search Tools Documents Help
*lab6.Nelia.report.md x lab6-3.asm x variant.asm x *lab6-4.asm x
%include "in_out.asm"
SECTION .data
msg: DB 'Введите значение x: ',0
div: DB 'результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call read
mov eax, x ; EAX: x
call atoi
mov ebx, 8 ; EBX: 8
mul ebx ; EAX: EAX * EBX
add eax, 6 ; EAX: EAX + ( 6)
xor edx, edx ; обнуляем EDX для корректной работы div
mov ebx, 2 ; EBX: 2
div ebx ; EAX: EAX / 2, EDX: остаток от деления
mov edi, eax ; запись результата вычисления в 'edi'
mov eax, div ; вызов подпрограммы печати
call sprintf ; сообщения 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprintf ; из 'edi' в виде символов
mov eax, rem ; вызов подпрограммы печати
call sprintf ; сообщения 'Остаток от деления: '
mov eax, edx ; вызов подпрограммы печати значения
call iprintf ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 3.21: Рис 21

Я создам и запущу исполняемый файл. Когда вы вводите значение 1, выход 1(рис 22)

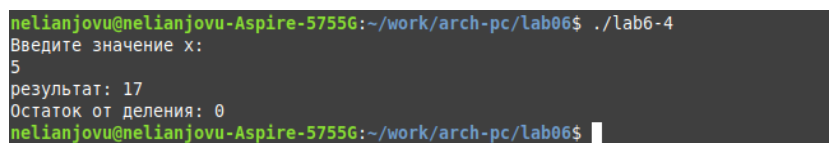


```
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ./lab6-4
Введите значение x:
1
результат: 1
Остаток от деления: 0
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$
```

Рис. 3.22: Рис 22

Я запускаю исполняемый файл еще раз, чтобы проверить работу программы с другим входным значением, на этот раз я буду использовать 5, а на выходе

должно быть 17. Программа работала корректно(рис 23)



```
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$ ./lab6-4
Введите значение x:
5
результат: 17
Остаток от деления: 0
nelianjovu@nelianjovu-Aspire-5755G:~/work/arch-pc/lab06$
```

Рис. 3.23: Рис 23

Листинг 3.1. Программа для вычисления значения выражения $(8x - 6)/2^*$

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите значение x: ',0
div: DB 'результат: ',0
rem: DB 'Остаток от деления: ',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 ; EAX=x
call atoi
mov ebx,8 ; EBX=8
mul ebx ; EAX=EAX*EBX
add eax,-6 ; EAX=EAX+(-6)
xor edx,edx ; обнуляем EDX для корректной работы div
```

```
mov ebx,2 ; EBX=2
div ebx ; EAX=EAX/2, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
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,edx ; 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
```

4 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

Список литературы

1. Архитектура ЭВМ
2. Таблица ASCII