

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Студент: Сухова Арина Евгеньевна

Группа: НПИбд-02-25

МОСКВА

2025 г.

Содержание

1. Цель работы.....	2
2. Задание.....	2
3. Выполнение лабораторной работы.....	2
3.1 Порядок выполнения лабораторной работы.....	2
3.2 Выполнение арифметических операций в NASM.....	6
3.3 Ответы на вопросы.....	9
3.4 Задание для самостоятельной работы.....	10
4. Выводы.....	12

1. Цель работы

Освоить арифметические инструкции языка ассемблера NASM и написать программы для вычисления арифметических выражений с неизвестной.

2. Задание

Написать программы для решения выражений.

3. Выполнение лабораторной работы

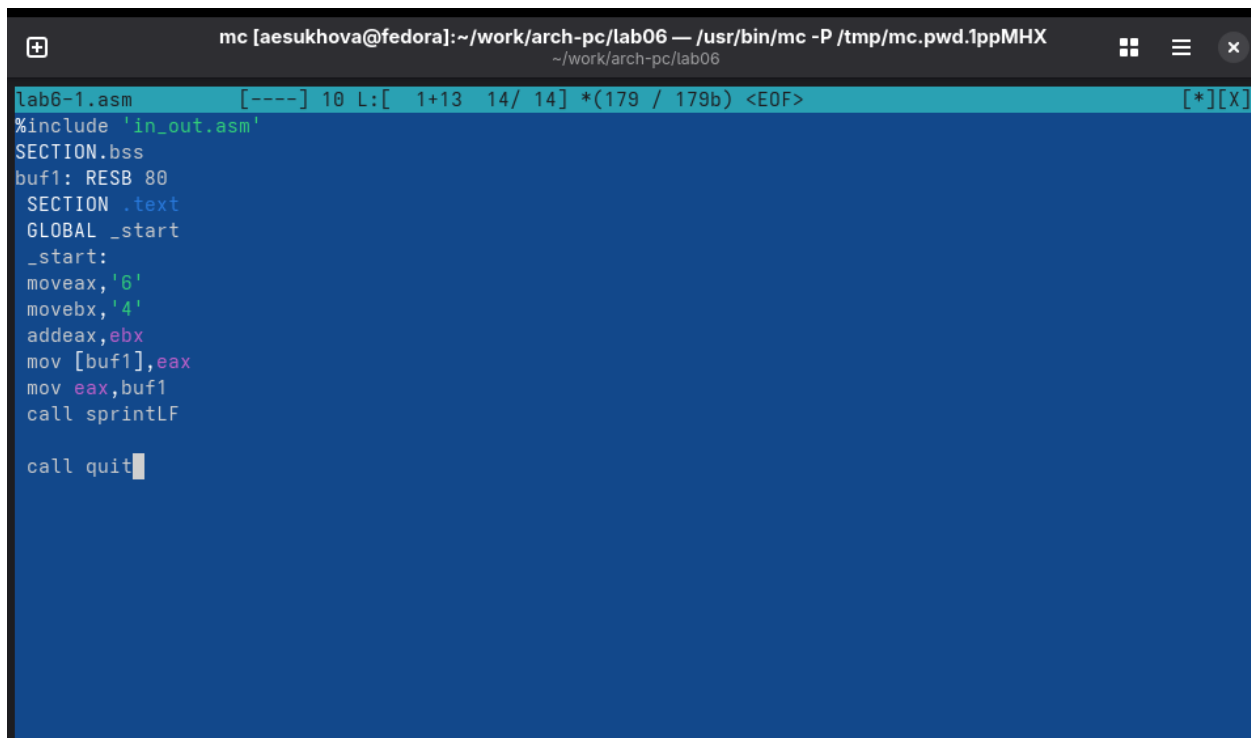
3.1 Порядок выполнения лабораторной работы

Создаем каталог для программ ЛБ6 и в нем создаем файл (рис.1)

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

Рис 1. Создаем каталог с помощью команды mkdir и файл с помощью команды touch

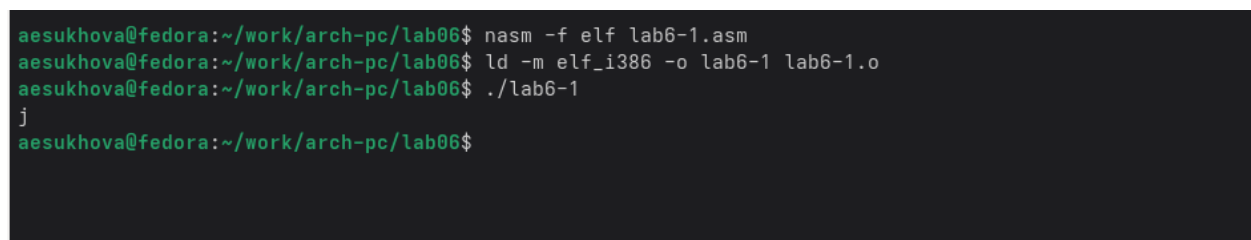
Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (рис. 2)

A screenshot of a microcontroller IDE window. The title bar shows the path `mc [aesukhova@fedora]:~/work/arch-pc/lab06` and the file `lab6-1.asm`. The code editor contains the following assembly code:

```
lab6-1.asm [----] 10 L:[ 1+13 14/ 14] *(179 / 179b) <EOF>
#include 'in_out.asm'
SECTION.bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
moveax,'6'
movebx,'4'
addeax,ebx
mov [buf1],eax
mov eax,buf1
call sprintf
call quit
```

Рис. 2 Заполняем файл

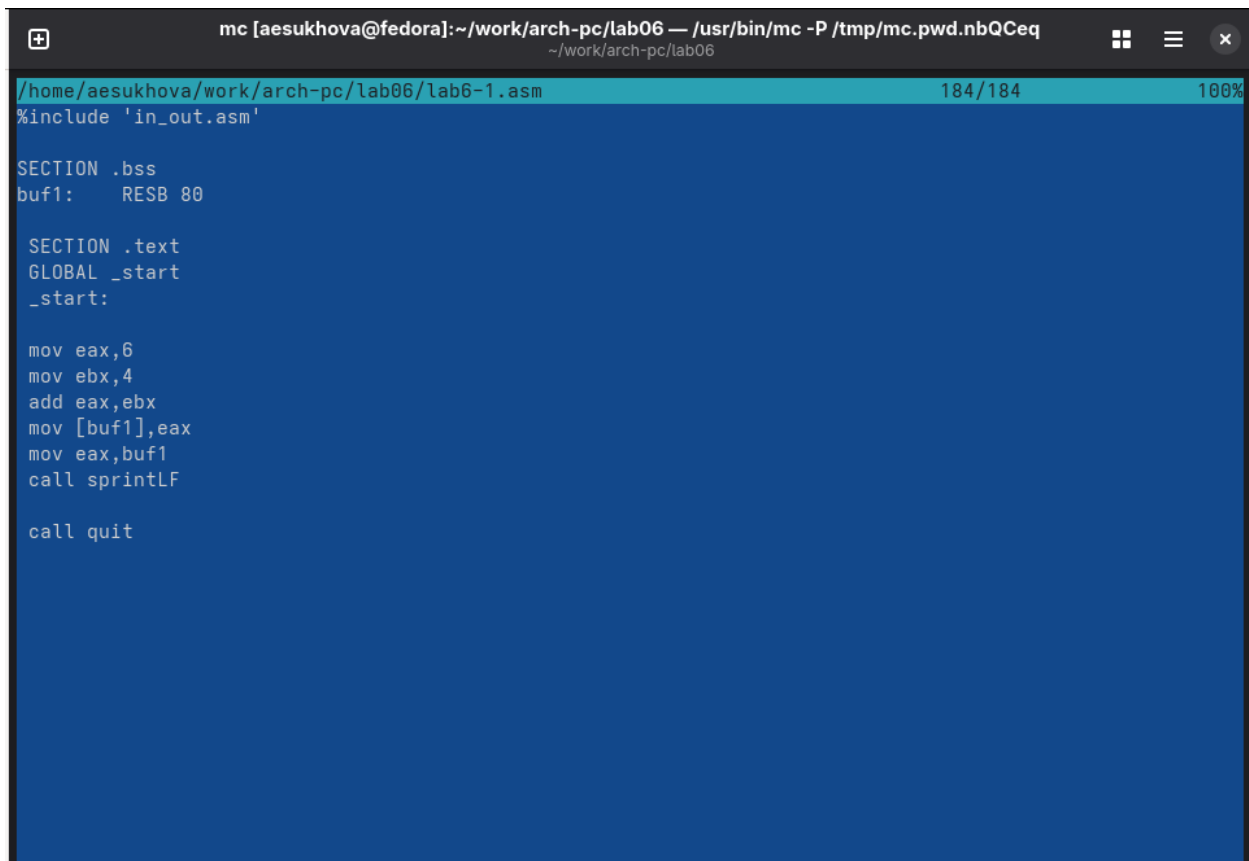
Создаем исполняемый файл и запускаем его (рис. 3)

A screenshot of a terminal window showing the compilation and execution of the assembly file. The commands and their outputs are:

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

Рис. 3 Запускаем файл и смотрим его работу

Открываем файл для редактирования и убираем кавычки с числовых значений (рис. 4)



The screenshot shows a window titled "mc [aesukhova@fedora]:~/work/arch-pc/lab06 — /usr/bin/mc -P /tmp/mc.pwd.nbQCeq". The file path in the title bar is "~/work/arch-pc/lab06". The editor displays the assembly file "lab6-1.asm" with 184/184 lines and 100% zoom. The code includes a header file "in_out.asm". It defines a .bss section for "buf1" with a size of 80 bytes. Then it defines a .text section starting at "_start". The code sets up registers: "mov eax,6", "mov ebx,4", "add eax,ebx", "mov [buf1],eax", "mov eax,buf1", and "call sprintLF". Finally, it calls "quit".

```
%include 'in_out.asm'

SECTION .bss
buf1:   RESB 80

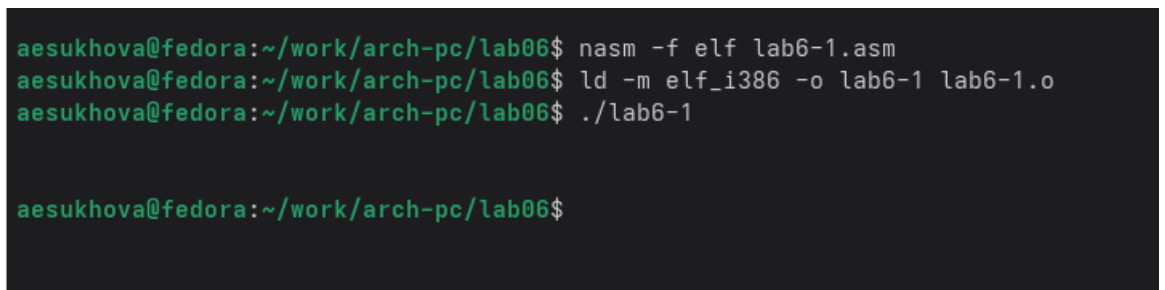
SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 4 Изменяем файл

Создаем исполняемый файл и запускаем его (рис.5)



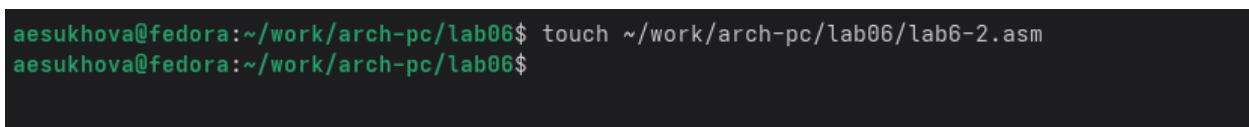
The screenshot shows a terminal window with the following commands and output:

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

aesukhova@fedora:~/work/arch-pc/lab06$
```

Рис. 5 Запускаем файл и смотрим на его работу

Создаем новый файл в каталоге (рис.6)

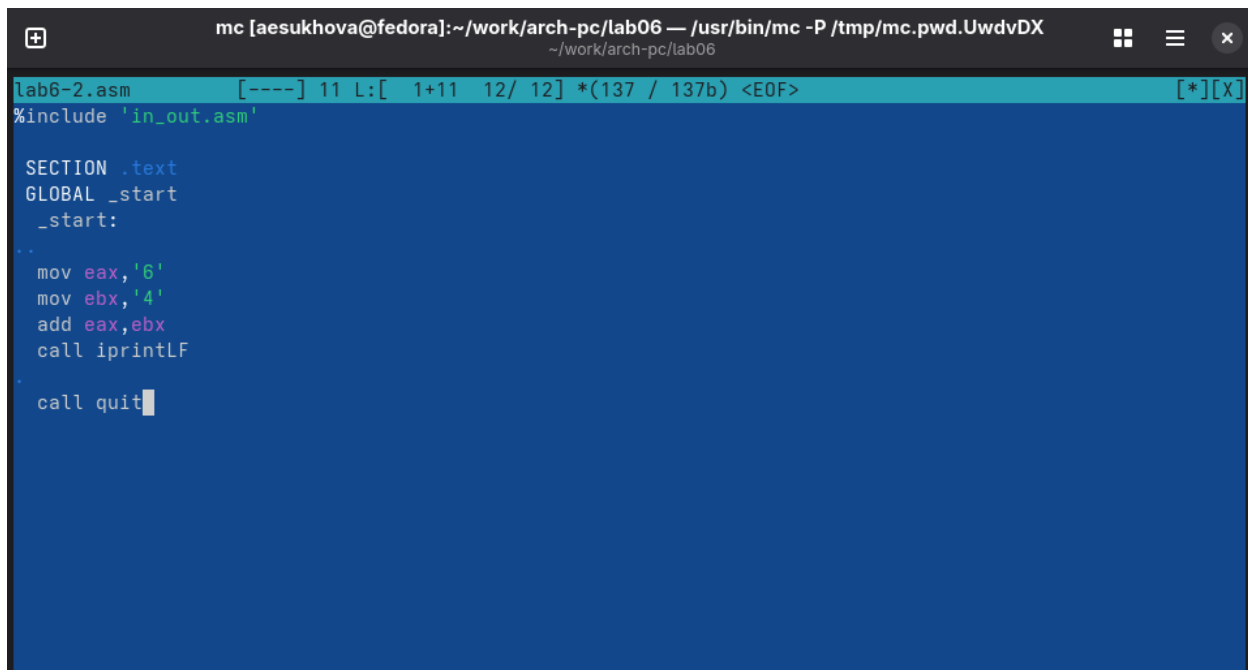


The screenshot shows a terminal window with the following commands:

```
aesukhova@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
aesukhova@fedora:~/work/arch-pc/lab06$
```

Рис. 6 Создаем файл

Заполняем файл в соответствии с листингом 6.2 (рис.7)



```
mc [aesukhova@fedora]:~/work/arch-pc/lab06 — /usr/bin/mc -P /tmp/mc.pwd.UwdvDX
~/work/arch-pc/lab06
lab6-2.asm [----] 11 L:[ 1+11 12/ 12] *(137 / 137b) <EOF> [*][X]
#include 'in_out.asm'

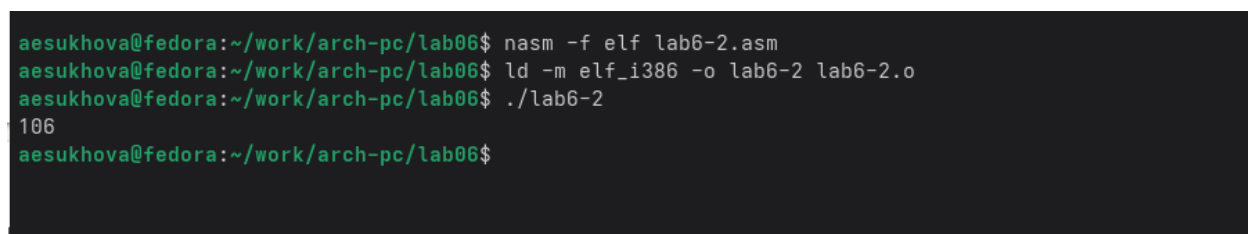
SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 7 Заполняем файл

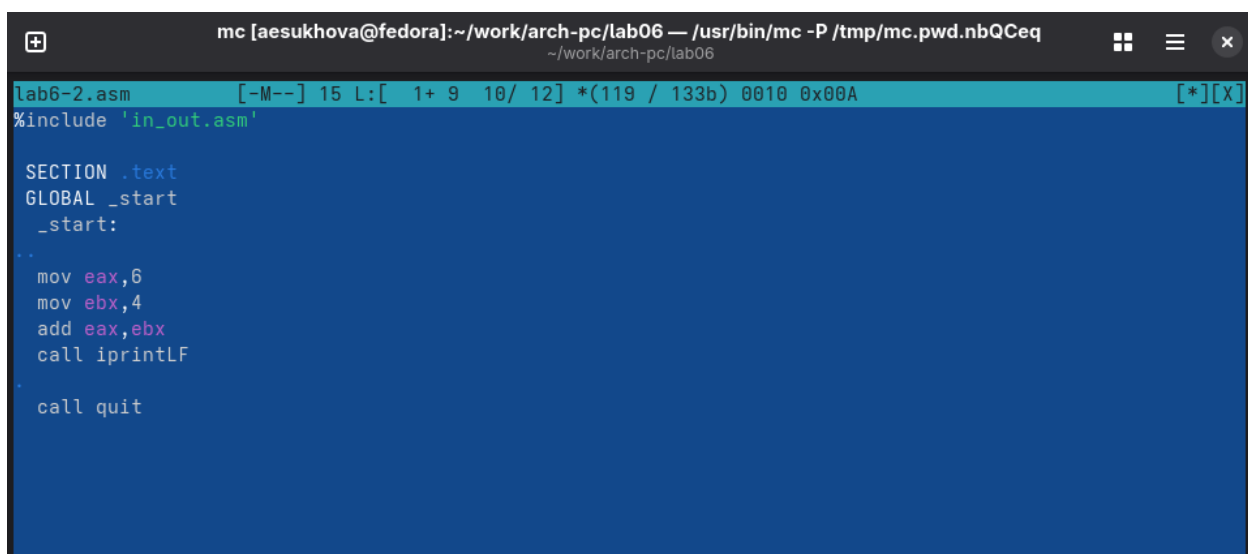
Создаем исполняемый файл и запускаем его (рис.8)



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

Рис. 8 Смотрим на работу программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис. 9)



```
mc [aesukhova@fedora]:~/work/arch-pc/lab06 — /usr/bin/mc -P /tmp/mc.pwd.nbQCeq
~/work/arch-pc/lab06
lab6-2.asm [-M--] 15 L:[ 1+ 9 10/ 12] *(119 / 133b) 0010 0x00A [*][X]
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 9 Изменяем файл

Создаем исполняемый файл и запускаем его (рис. 10)

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

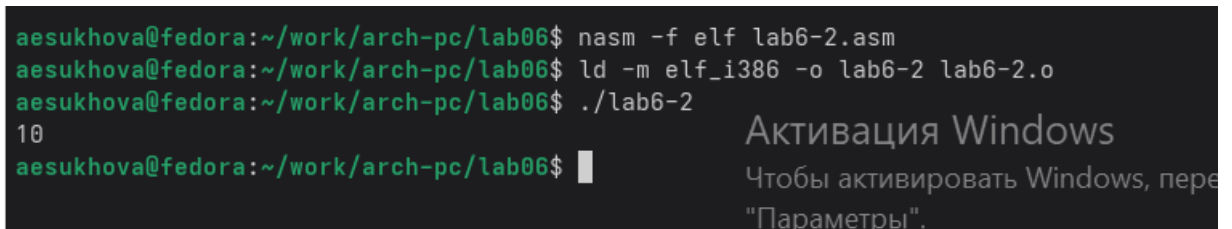
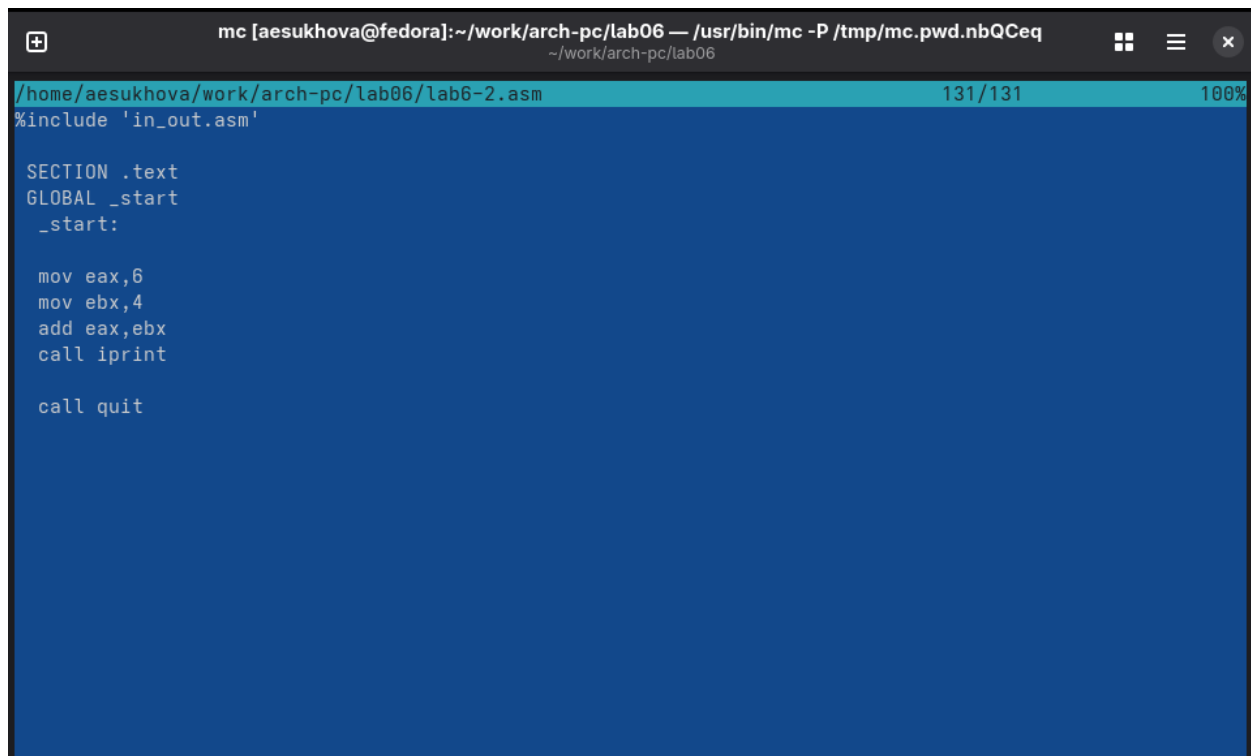


Рис. 10 Смотрим на работу программы

Снова открываем файл для редактирования и меняем `iprintLF` на `iprint` (рис. 11)



```
mc [aesukhova@fedora]:~/work/arch-pc/lab06 — /usr/bin/mc -P /tmp/mc.pwd.nbQCeq
~/work/arch-pc/lab06

/home/aesukhova/work/arch-pc/lab06/lab6-2.asm 131/131 100%
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

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

    call quit
```

Рис. 11 Изменяем файл

Создаем исполняемый файл и запускаем его (рис.12)

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

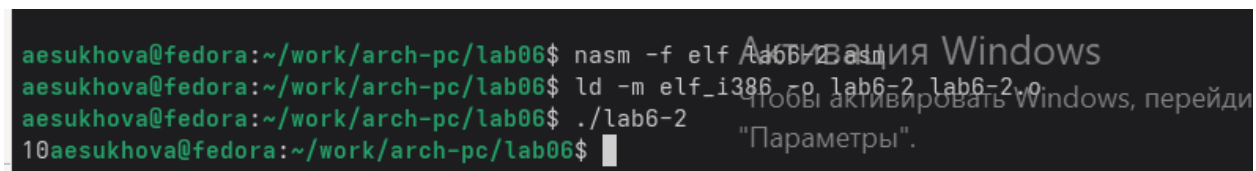


Рис. 12 Смотрим, как сработала программа

Вывод функций `iprintLF` и `iprint` отличаются только тем, что `LF` переносит на новую строку.

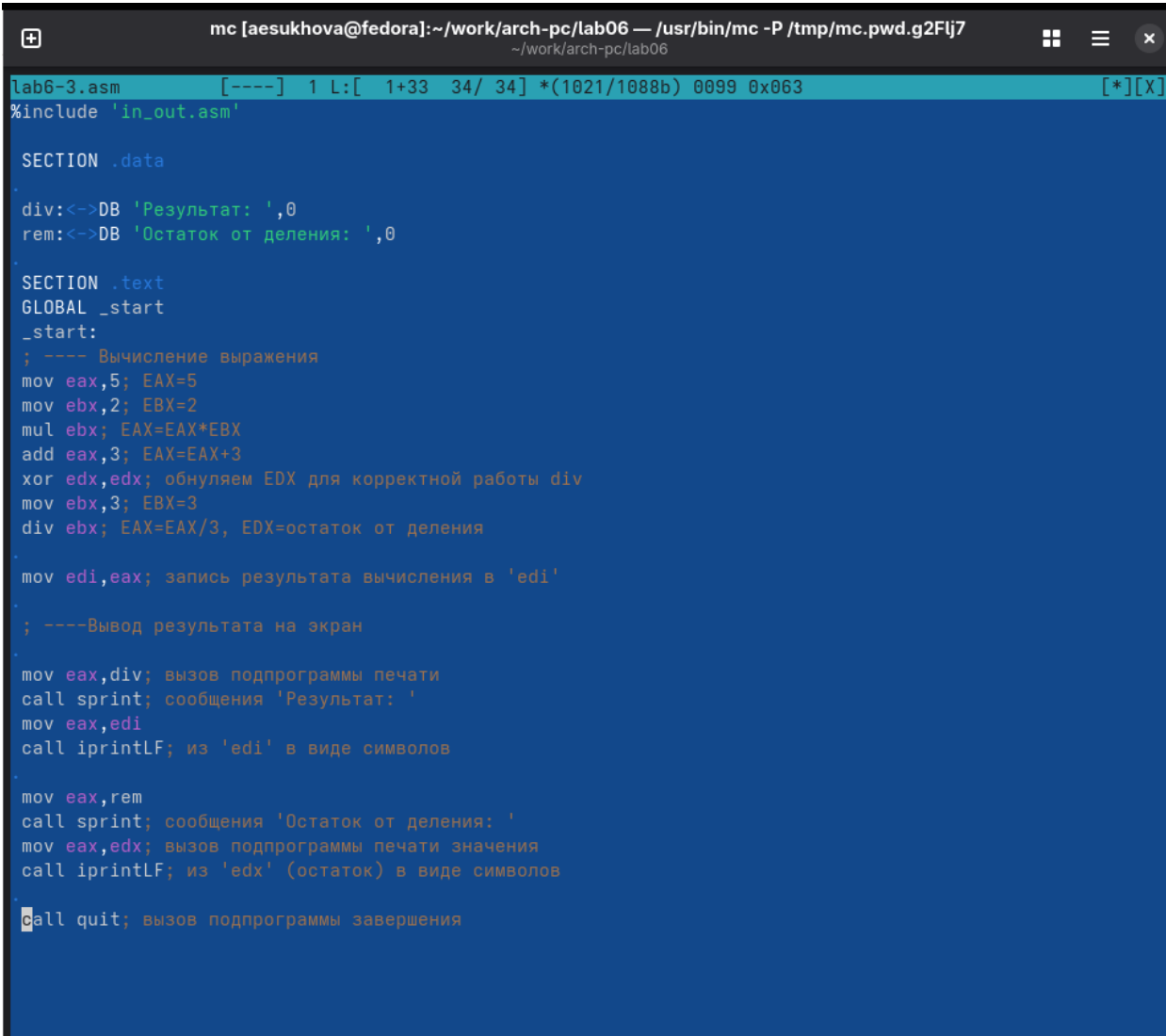
3.2 Выполнение арифметических операций в NASM

Создаем новый файл в каталоге (рис. 13)

```
10 aesukhova@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
aesukhova@fedora:~/work/arch-pc/lab06$
```

Рис. 13 Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.3 (рис. 14)



```
mc [aesukhova@fedora]:~/work/arch-pc/lab06 — /usr/bin/mc -P /tmp/mc.pwd.g2Flj7
~/work/arch-pc/lab06
lab6-3.asm [----] 1 L: [ 1+33 34/ 34] *(1021/1088b) 0099 0x063 [*][X]
%include 'in_out.asm'

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
call iprintLF; из 'edi' в виде символов

mov eax,rem
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
```

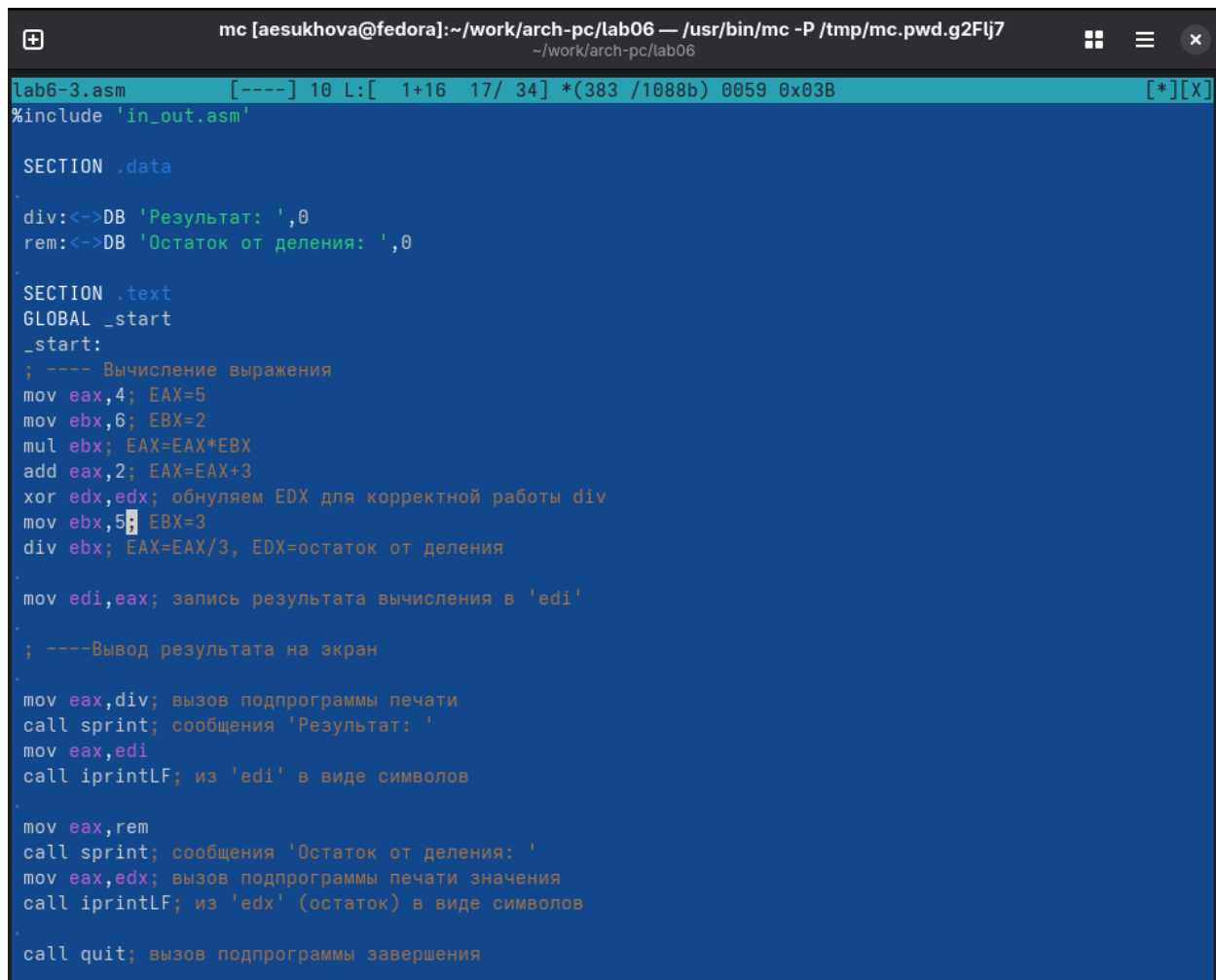
Рис. 14 Заполняем файл

Создаем исполняемый файл и запускаем его (рис.15)

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

Рис. 15 Смотрим на результат работы программы

Открываем файл и редактируем его для вычисления выражения $f(x) = (4 * 6 + 2)/5$ (рис. 16)



```
mc [aesukhova@fedora]:~/work/arch-pc/lab06 — /usr/bin/mc -P /tmp/mc.pwd.g2Flj7
~/work/arch-pc/lab06
lab6-3.asm [----] 10 L:[ 1+16 17/ 34] *(383 /1088b) 0059 0x03B [*][X]
#include 'in_out.asm'

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

SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4; EAX=5
mov ebx,6; EBX=2
mul ebx; EAX=EAX*EBX
add eax,2; EAX=EAX+3
xor edx,edx; обнуляем EDX для корректной работы div
mov ebx,5; EBX=3
div ebx; EAX=EAX/3, EDX=остаток от деления

mov edi,eax; запись результата вычисления в 'edi'

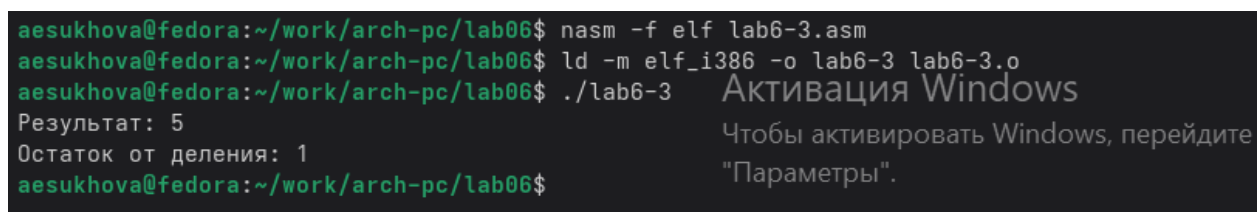
; ----Вывод результата на экран
mov eax,div; вызов подпрограммы печати
call sprint; сообщения 'Результат: '
mov eax,edi
call iprintLF; из 'edi' в виде символов

mov eax,rem
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. 16 Редактируем файл

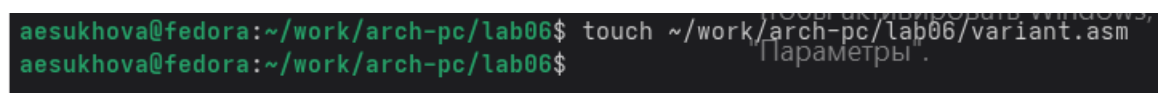
Транслируем файл и запускаем программу (рис. 17)



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

Рис. 17 Проверяем правильность написания программы

Создаем новый файл в каталоге (рис.18)



```
aesukhova@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
aesukhova@fedora:~/work/arch-pc/lab06$
```

Рис. 18 Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.4 (рис. 19)


```
mc [aesukhova@fedora]:~/work/arch-pc/lab06 — /usr/bin/mc -P /tmp/mc.pwd.SlCwlY
~/work/arch-pc/lab06
variant.asm [-M--] 1 L:[ 1+33 34/ 34] *(417 / 426b) 0099 0x063 [*][X]
#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
```

Рис. 19 Заполняем файл

Транслируем файл и запускаем программу (рис. 20)

```
aesukhova@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
aesukhova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
aesukhova@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032253644
Ваш вариант: 5
aesukhova@fedora:~/work/arch-pc/lab06$
```

Активация Windows
Чтобы активировать Windows, перейдите в "Параметры".

Рис. 20 Проверяем результат программы

3.3 Ответы на вопросы

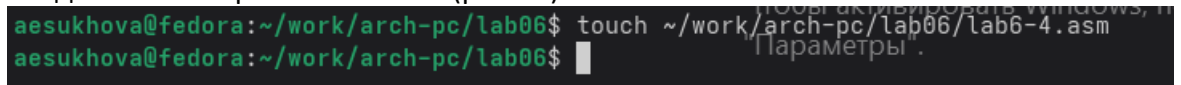
1. Строка "mov eax,rem" и строка "call sprintf" отвечают за вывод на экран сообщения 'Ваш вариант:'.
2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре ecx, а количество символов в строке (максимальное количество символов, которое может быть считано)

сохраняется в регистре `edx`. Затем вызывается процедура `sread`, которая выполняет чтение строки.

3. Инструкция `"call atoi"` используется для преобразования строки в целое число. Она принимает адрес строки в регистре `eax` и возвращает полученное число в регистре `eax`.
4. 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. Остаток от деления записывается в регистр `edx`.
6. Инструкция `"inc edx"` используется для увеличения значения в регистре `edx` на 1. В данном случае, она увеличивает остаток от деления на 1.
7. 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.

3.4 Задание для самостоятельной работы

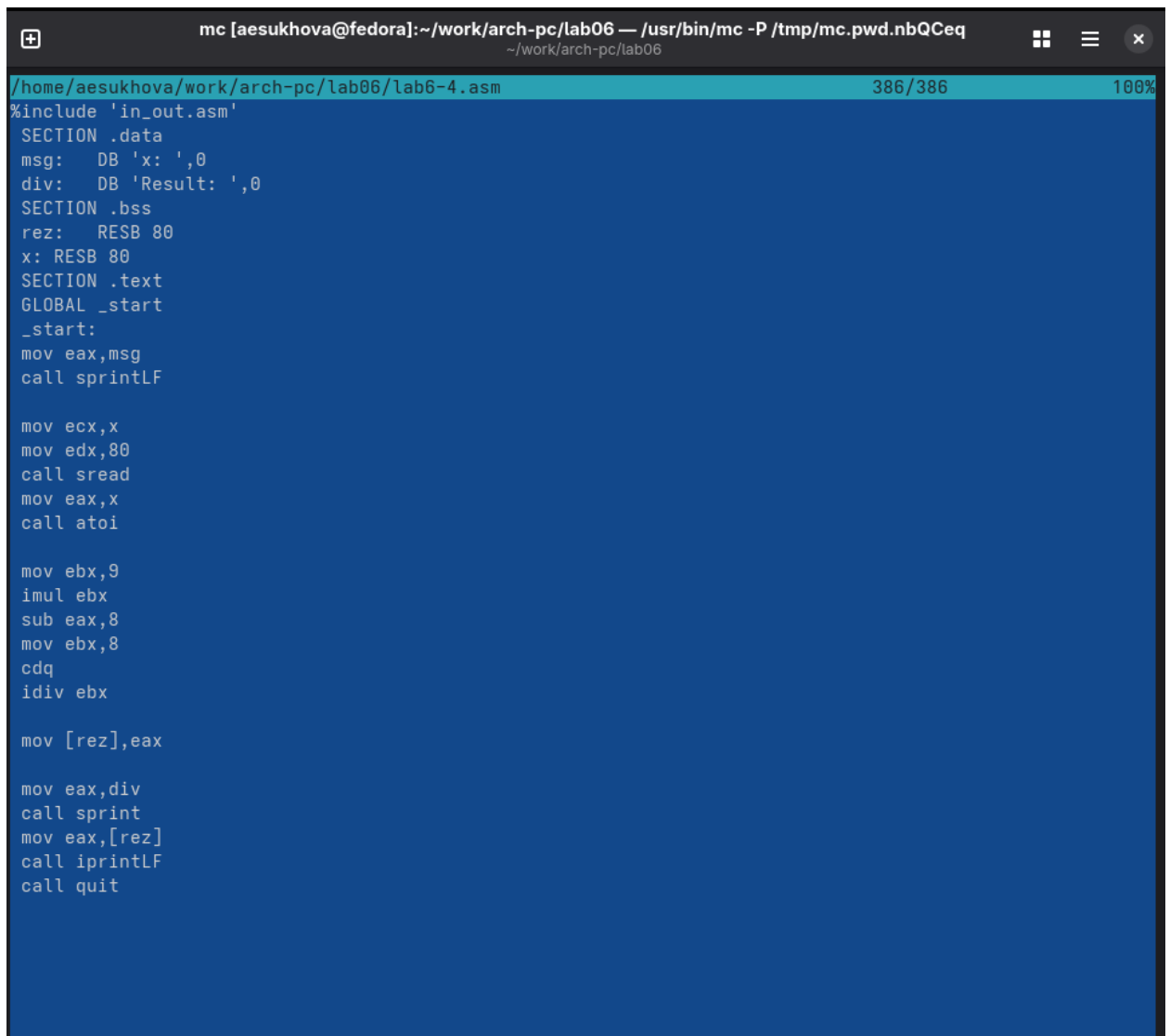
Создаем новый файл в каталоге (рис.21)



```
aesukhova@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-4.asm
aesukhova@fedora:~/work/arch-pc/lab06$
```

Рис. 21 Создаем файл

Открываем его и заполняем, чтобы решалось выражение $f(x) = (9x - 8)/8$ (рис. 22)



```
mc [aesukhova@fedora]:~/work/arch-pc/lab06 — /usr/bin/mc -P /tmp/mc.pwd.nbQCeq
~/work/arch-pc/lab06
/home/aesukhova/work/arch-pc/lab06/lab6-4.asm 386/386 100%
#include 'in_out.asm'
SECTION .data
msg: DB 'x: ',0
div: DB 'Result: ',0
SECTION .bss
rez: RESB 80
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

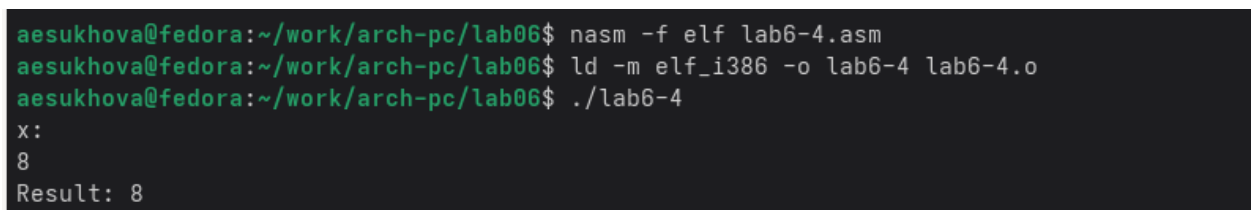
mov ebx,9
imul ebx
sub eax,8
mov ebx,8
cdq
idiv ebx

mov [rez],eax

mov eax,div
call sprintf
mov eax,[rez]
call iprintLF
call quit
```

Рис. 22 Заполняем файл

Компилируем программу и проверяем для $x_1 = 8$ (рис. 23)



```
aesukhova@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
aesukhova@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
aesukhova@fedora:~/work/arch-pc/lab06$ ./lab6-4
x:
8
Result: 8
```

Рис. 23 Проверяем работу программы

Компилируем программу и проверяем для $x = 64$ (рис. 24)

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

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

Рис. 24 Проверяем работу программы

4. Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.