

Отчёта по лабораторной работе №6

Арифметические операции в NASM.

Гурбанов Сарча

Содержание

1	<u>Цель работы</u>	4
2	<u>Задание</u>	5
3	<u>Выполнение лабораторной работы</u>	6
1.	<u>Символьные и численные данные в NASM</u>	6
2.	<u>Выполнение арифметических операций в NASM</u>	11
3.	<u>Ответы на вопросы по программе</u>	14
4.	<u>Задание для самостоятельной работы</u>	15
4	<u>Выводы</u>	18

Список иллюстраций

1.	<u>Создаем каталог с помощью команды mkdir и файл с помощью команды touch</u>	6
2.	<u>Заполняем файл</u>	7
3.	<u>Запускаем файл и смотрим на его работу</u>	7
4.	<u>Изменяем файл</u>	8
5.	<u>Запускаем файл и смотрим на его работу</u>	8
6.	<u>Создаем файл</u>	8
7.	<u>Заполняем файл</u>	9
8.	<u>Смотрим на работу программы</u>	9
9.	<u>Изменяем файл</u>	10
10.	<u>Смотрим на работу программы</u>	10
11.	<u>Изменяем файл</u>	11
12.	<u>Смотрим на работу программы</u>	11
13.	<u>Создаем файл</u>	11
14.	<u>Заполняем файл</u>	12
15.	<u>Смотрим на результат работы программы</u>	12
16.	<u>Редактируем файл</u>	13
17.	<u>Смотрим на результат работы программы</u>	13
18.	<u>Создаем файл</u>	13
19.	<u>Заполняем файл</u>	14
20.	<u>Проверяем результат работы программы</u>	14
21.	<u>Создаем файл</u>	15
22.	<u>Заполняем файл</u>	16
23.	<u>Проверяем работу программы</u>	16
24.	<u>Проверяем работу программы</u>	17

1 Цель работы

Освоить арифметических инструкций 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 Символьные и численные данные в NASM

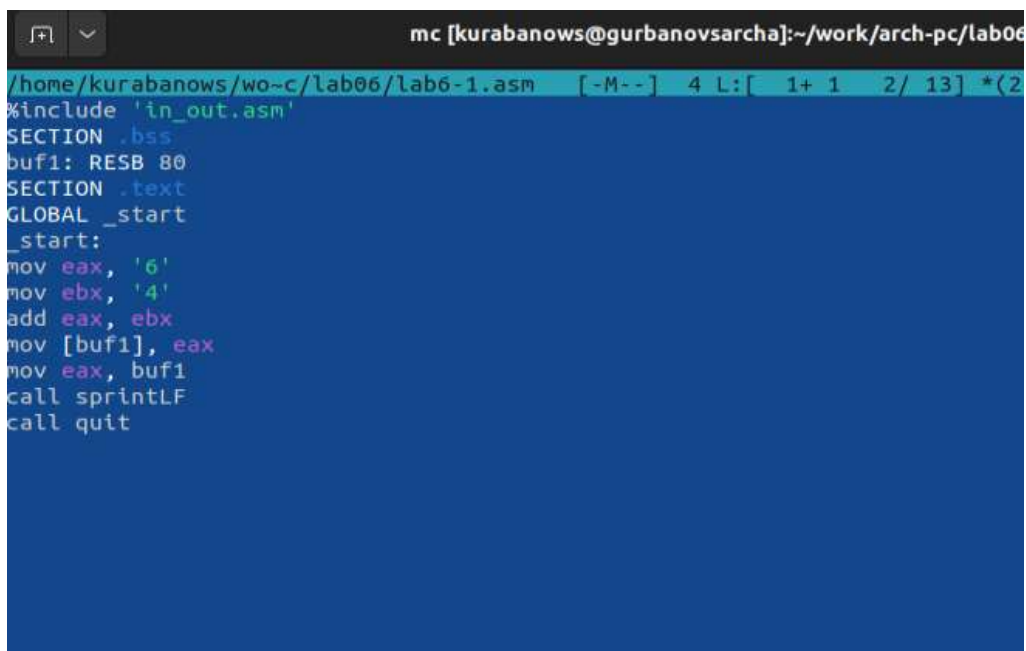
Создаем каталог для программ ЛБ6, и в нем создаем файл (рис. [3.1](#)).



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

Рис. 3.1: Создаем каталог с помощью команды `mkdir` и файл с помощью команды `touch`

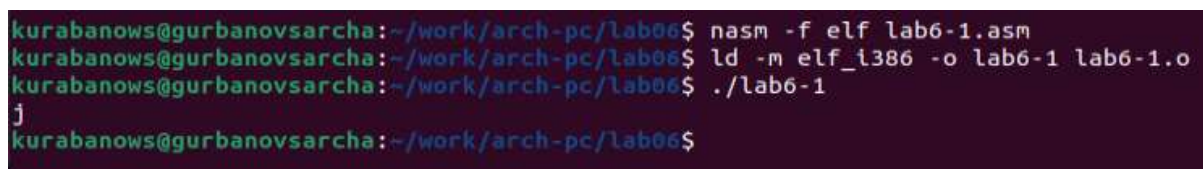
Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (рис. [3.2](#)).

A screenshot of a text editor window titled 'mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab06'. The editor shows the contents of a file named '/home/kurabanows/work/arch-pc/lab06/lab6-1.asm'. The code is as follows:

```
%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
```

Рис. 3.2: Заполняем файл

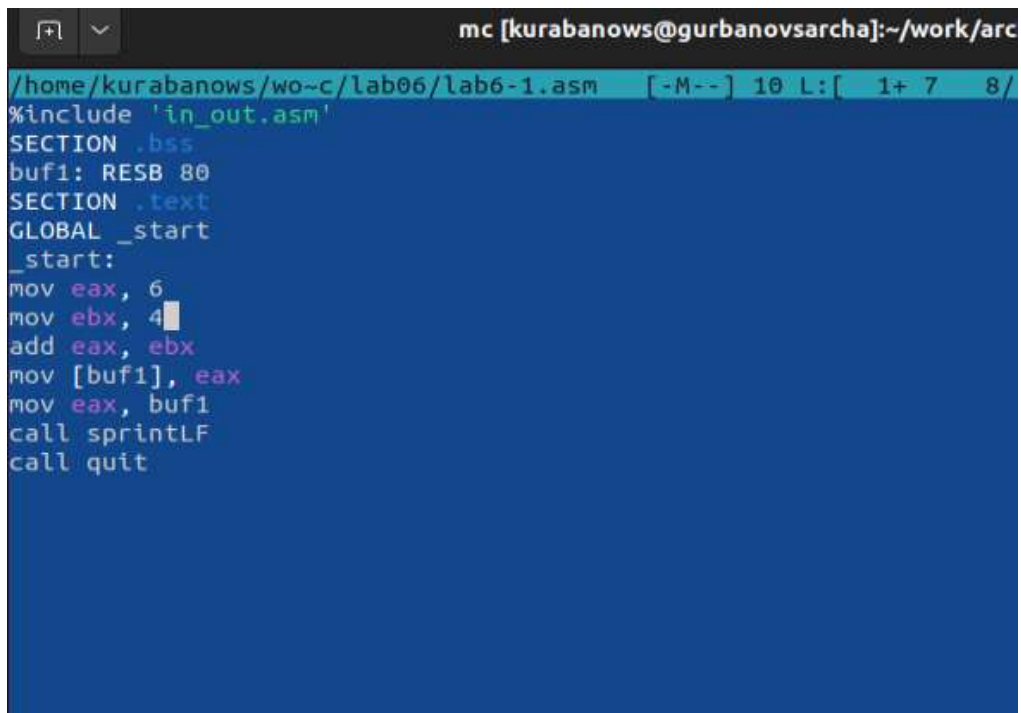
Создаем исполняемый файл и запускаем его (рис. [3.3](#)).

A screenshot of a terminal window showing the following commands and output:

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

Рис. 3.3: Запускаем файл и смотрим на его работу

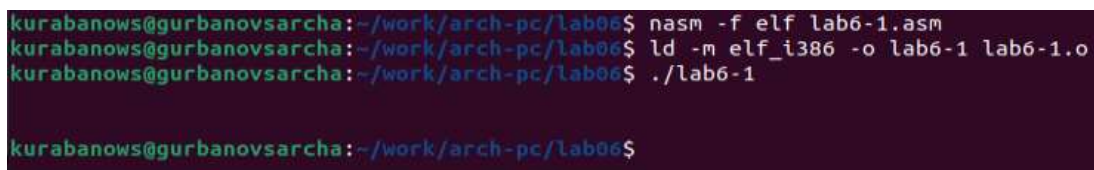
Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис. [3.4](#)).



```
mc [kurabanows@gurbanovsarcha]:~/work/arc
/home/kurabanows/wo~c/lab06/lab6-1.asm [-M--] 10 L: [ 1+ 7 8/
%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
```

Рис. 3.4: Изменяем файл

Создаем исполняемый файл и запускаем его (рис. 3.5).



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

kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$
```

Рис. 3.5: Запускаем файл и смотрим на его работу

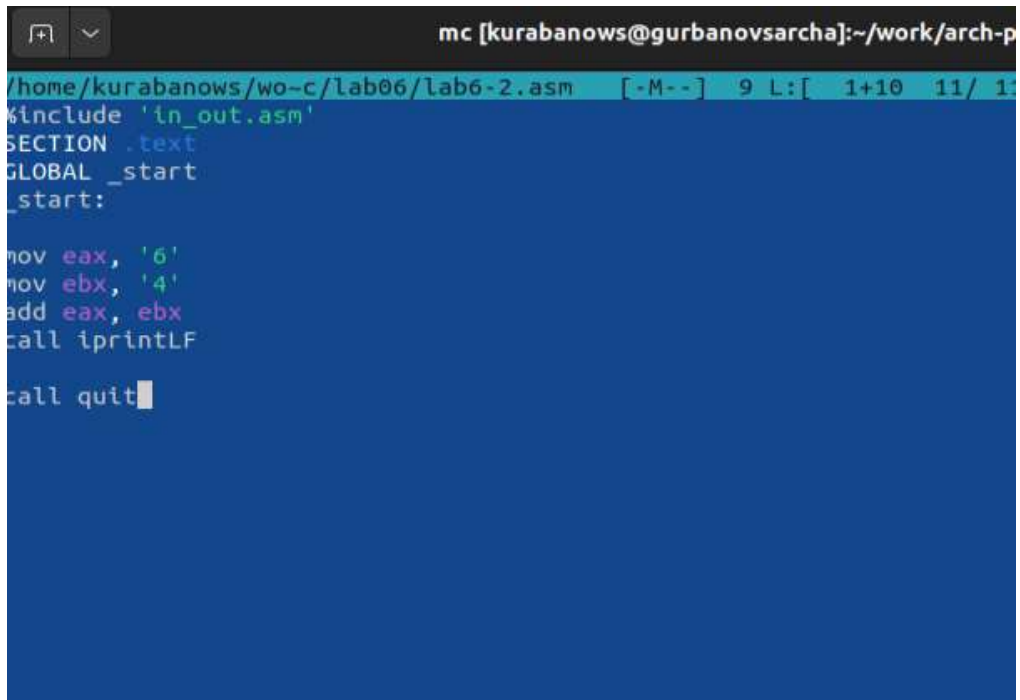
Создаем новый файл в каталоге (рис. 3.6).



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

Рис. 3.6: Создаем файл

Заполняем файл в соответствии с листингом 6.2 (рис. 3.7).



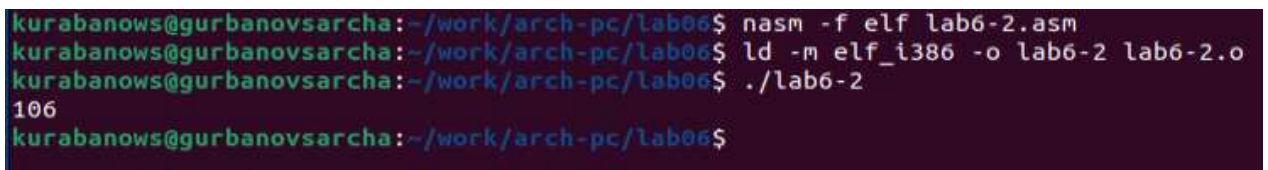
```
mc [kurabanows@gurbanovsarcha]:~/work/arch-p
/home/kurabanows/work/lab06/lab6-2.asm [-M--] 9 L: [ 1+10 11/ 1:
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 3.7: Заполняем файл

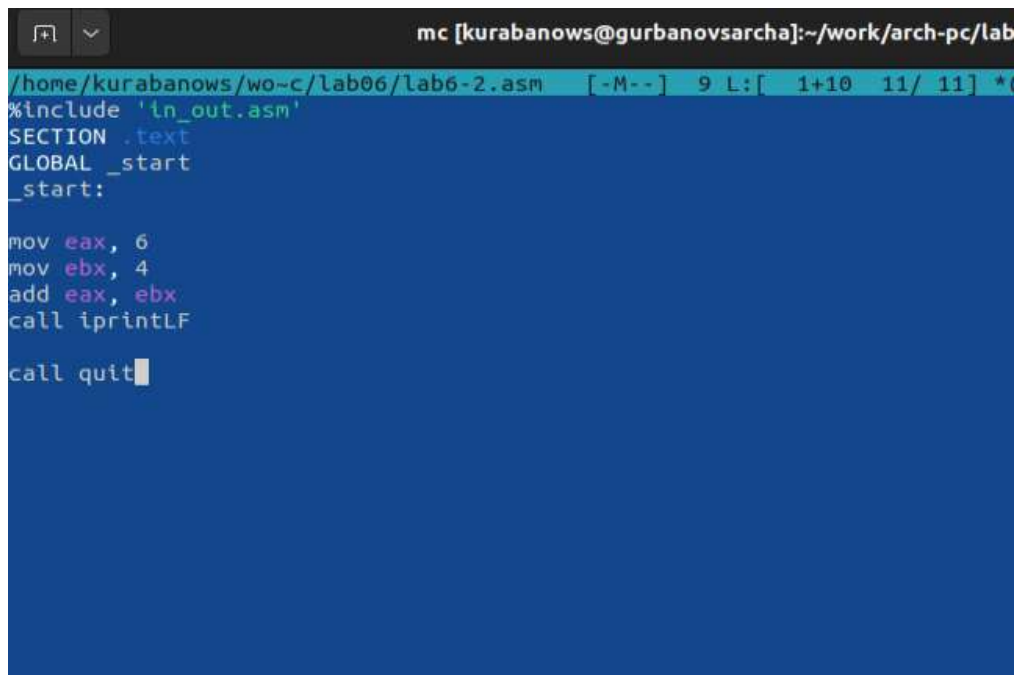
Создаем исполняемый файл и запускаем его (рис. 3.8).



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

Рис. 3.8: Смотрим на работу программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис. 3.9).

A screenshot of a text editor window titled 'mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab'. The editor shows the contents of the file '/home/kurabanows/work/arch-pc/lab06/lab6-2.asm'. The code is as follows:

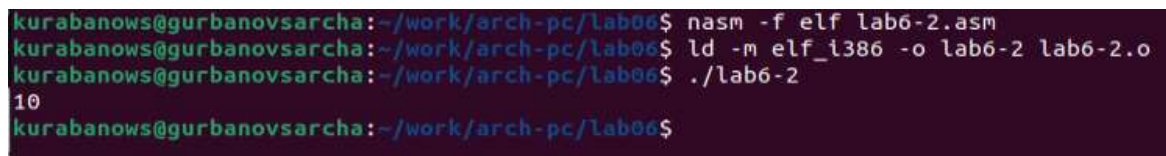
```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 3.9: Изменяем файл

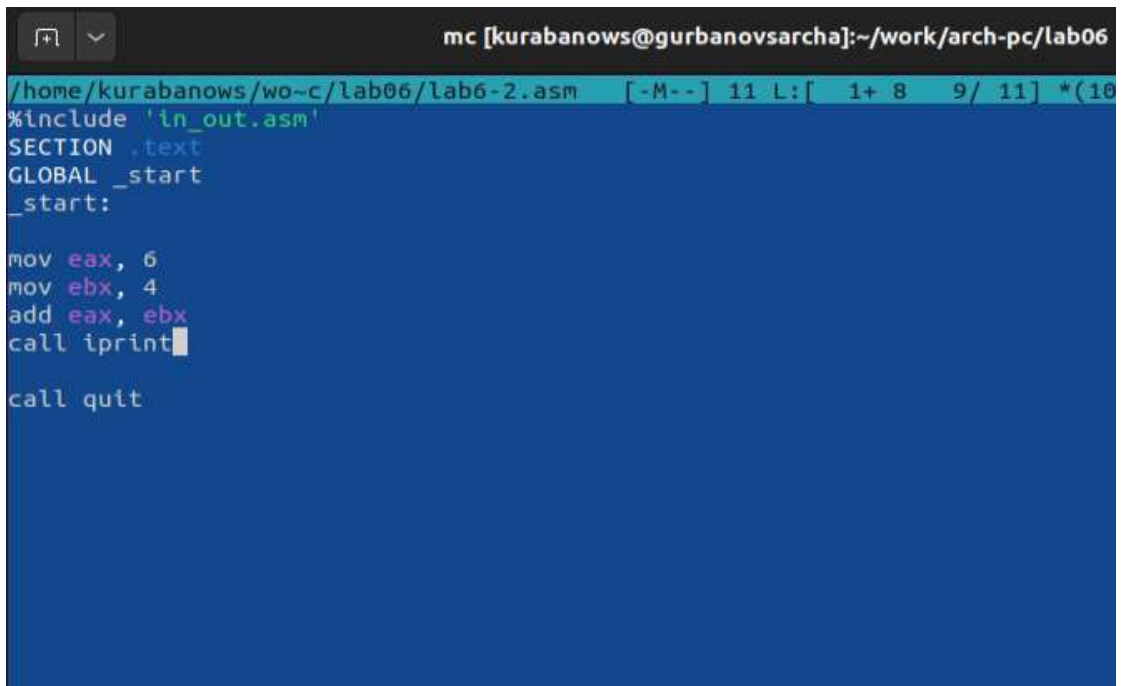
Создаем исполняемый файл и запускаем его (рис. [3.10](#)).

A screenshot of a terminal window showing the following commands and output:

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

Рис. 3.10: Смотрим на работу программы

Снова открываем файл для редактирования и меняем `iprintLF` на `iprint` (рис. [3.11](#)).

A screenshot of a code editor window. The title bar shows the path ~/work/arch-pc/lab06. The editor displays assembly code for a file named lab6-2.asm. The code includes a header, a section declaration, a global symbol, and several instructions: mov eax, 6; mov ebx, 4; add eax, ebx; call iprint; and call quit.

```
mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab06
/home/kurabanows/wo~c/lab06/lab6-2.asm [-M--] 11 L: [ 1+ 8 9/ 11] *(10
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

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

Рис. 3.11: Изменяем файл

Создаем исполняемый файл и запускаем его (рис. 3.12).

A screenshot of a terminal window showing the process of building and running the assembly program. The commands used are nasm to assemble the file, ld to link it into an executable, and ./lab6-2 to run it.

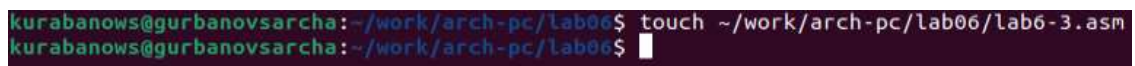
```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ ./lab6-2
10kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$
```

Рис. 3.12: Смотрим на работу программы

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

3.2 Выполнение арифметических операций в NASM

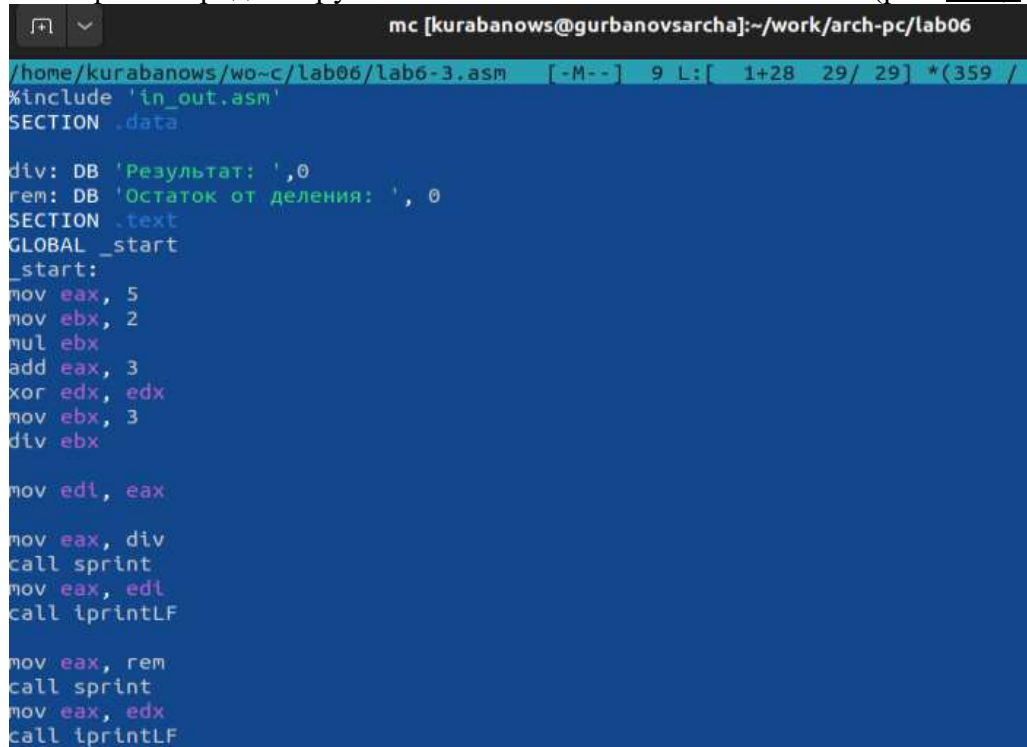
Создаем новый файл в каталоге (рис. 3.13).

A screenshot of a terminal window showing the command to create a new file named lab6-3.asm using the touch command.

```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$
```

Рис. 3.13: Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.3 (рис. 3.14).



```
mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab06
/home/kurabanows/work/arch-pc/lab06/lab6-3.asm [-M--] 9 L:[ 1+28 29/ 29] *(359 /
#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

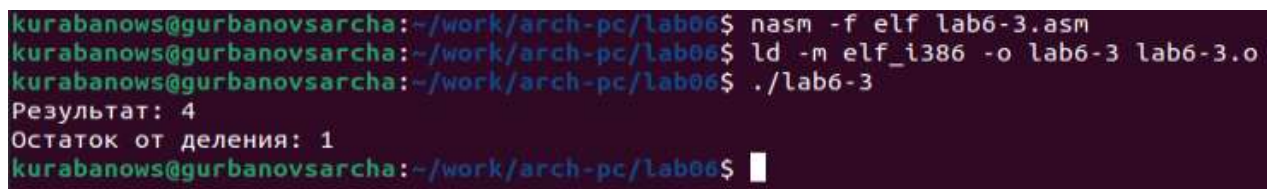
mov edi, eax

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

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

Рис. 3.14: Заполняем файл

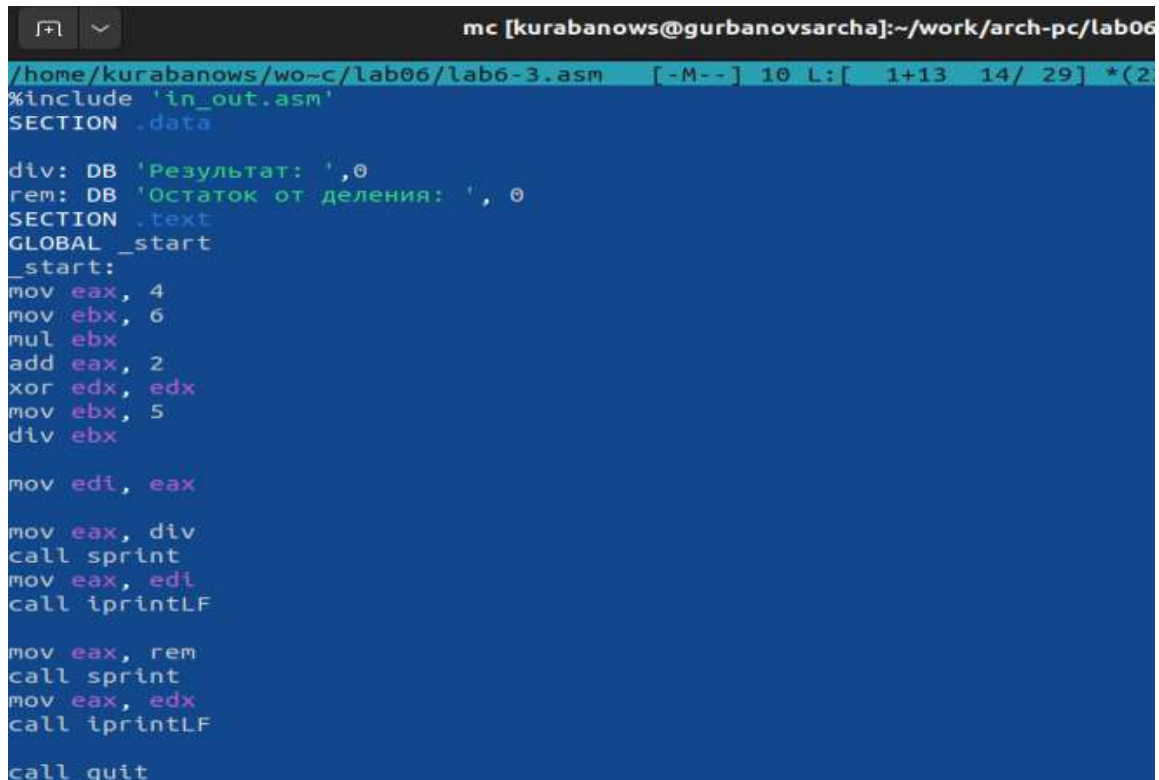
Создаем исполняемый файл и запускаем его (рис. 3.15).



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

Рис. 3.15: Смотрим на результат работы программы

Открываем файл и редактируем его для вычисления выражения $f(\heartsuit) = (4 \heartsuit + 2)/5$ (рис. 3.16).



```
mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab06
/home/kurabanows/work/arch-pc/lab06/lab6-3.asm [-M--] 10 L:[ 1+13 14/ 29] *(2
#include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ', 0
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
```

Рис. 3.16: Редактируем файл

Компилируем файл и запускаем программу (рис. [3.17](#)).



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

Рис. 3.17: Смотрим на результат работы программы

Создаем новый файл в каталоге (рис. [3.18](#)).



```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$
```

Рис. 3.18: Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.4 (рис. [3.19](#)).

```
mc [kurabanows@gurbanovsarcha]:~/work/arch-pc/lab06
/home/kurabanows/work/arch-pc/lab06/variant.asm [-M--] 7 L: [ 1+23 24/ 32] *(326 /
%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 sprintf
call quit
```

Рис. 3.19: Заполняем файл

Компилируем файл и запускаем его (рис. 3.20).

```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ nasm -f elf variant.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032244850
Ваш вариант: 11
```

Рис. 3.20: Проверяем результат работы программы

3.3 Ответы на вопросы по программе

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.

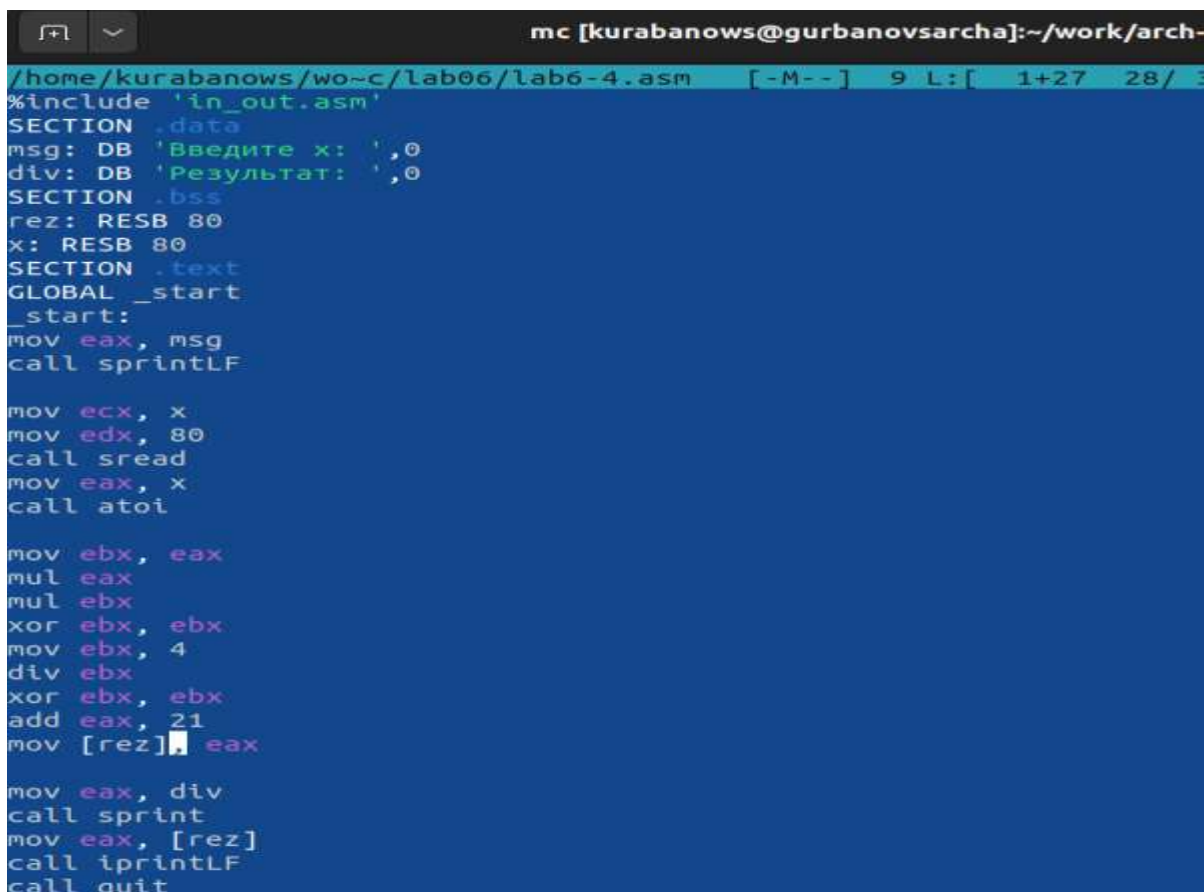
3.4 Задание для самостоятельной работы

Создаем новый файл в каталоге (рис. 3.21).

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

Рис. 3.21: Создаем файл

Открываем его и заполняем, чтобы решалось выражение $f(x) = x^3 - x^{1/3} + 21$ (рис. 3.22).



```
mc [kurabanows@gurbanovsarcha]:~/work/arch-  
/home/kurabanows/work/lab06/lab6-4.asm [-M--] 9 L:[ 1+27 28/ 3  
%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:  
mov eax, msg  
call sprintf  
  
mov ecx, x  
mov edx, 80  
call sread  
mov eax, x  
call atoi  
  
mov ebx, eax  
mul eax  
mul ebx  
xor ebx, ebx  
mov ebx, 4  
div ebx  
xor ebx, ebx  
add eax, 21  
mov [rez], eax  
  
mov eax, div  
call sprintf  
mov eax, [rez]  
call iprintf  
call quit
```

Рис. 3.22: Заполняем файл

Компилируем программу и проверяем для $x=1$ (рис. 3.23).



```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm  
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o  
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ ./lab6-4  
Введите x:  
1  
Результат: 21  
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$
```

Рис. 3.23: Проверяем работу программы

Компилируем программу и проверяем для $x=3$ (рис. 3.24).


```
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
3
Результат: 27
kurabanows@gurbanovsarcha:~/work/arch-pc/lab06$
```

Рис. 3.24: Проверяем работу программы

4 Выводы

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