

Отчёта по лабораторной работе №6

Арифметические операции в NASM.

Жозе Рамос Домингуш

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Символьные и численные данные в NASM	6
3.2	Выполнение арифметических операций в NASM	11
3.3	Ответы на вопросы по программе	14
3.4	Задание для самостоятельной работы	15
4	Выводы	17

Список иллюстраций

3.1	Создаем каталог с помощью команды <code>mkdir</code> и файл с помощью команды <code>touch</code>	6
3.2	Заполняем файл	7
3.3	Запускаем файл и смотрим на его работу	7
3.4	Изменяем файл	8
3.5	Запускаем файл и смотрим на его работу	8
3.6	Создаем файл	8
3.7	Заполняем файл	9
3.8	Смотрим на работу программы	9
3.9	Изменяем файл	10
3.10	Смотрим на работу программы	10
3.11	Изменяем файл	11
3.12	Смотрим на работу программы	11
3.13	Создаем файл	11
3.14	Заполняем файл	12
3.15	Смотрим на результат работы программы	12
3.16	Редактируем файл	13
3.17	Смотрим на результат работы программы	13
3.18	Создаем файл	13
3.19	Заполняем файл	14
3.20	Проверяем результат работы программы	14
3.21	Создаем файл	15
3.22	Заполняем файл	16
3.23	Проверяем работу программы	16
3.24	Проверяем работу программы	16

1 Цель работы

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

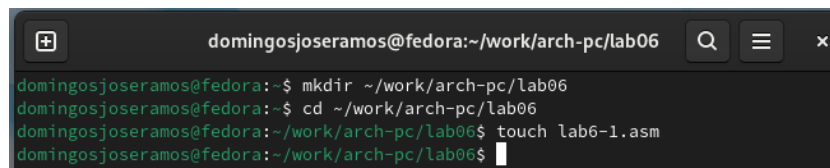
2 Задание

Написать программы для решения выражений.

3 Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

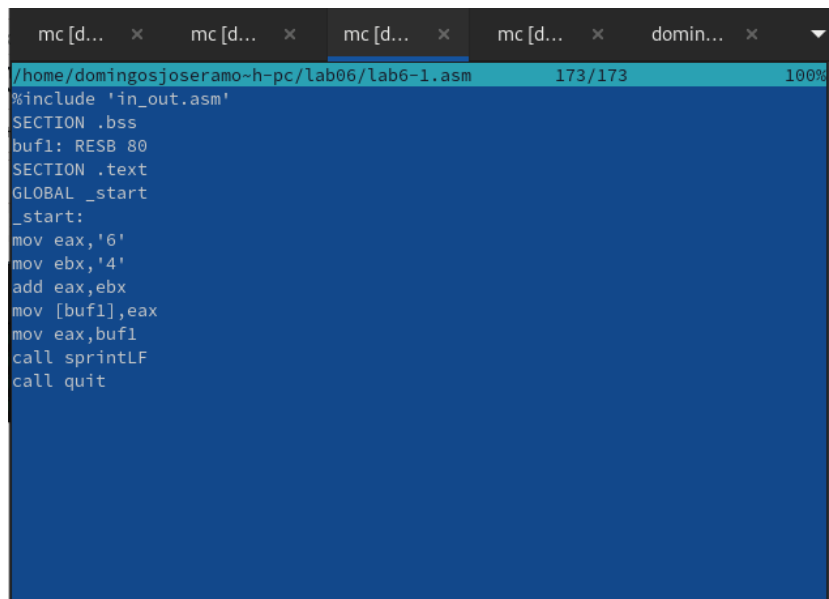
Создаем каталог для программ ЛБ6, и в нем создаем файл (рис. fig. 3.1).

A terminal window with a dark background. The title bar shows the user 'domingosjoseramos' on a 'fedora' machine in the directory '~/work/arch-pc/lab06'. The terminal contains the following commands and their outputs:

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

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

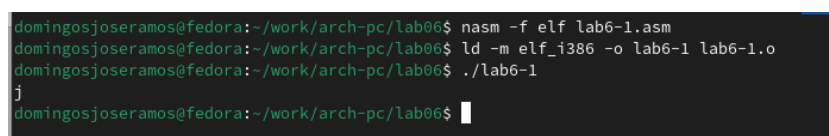
Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (рис. fig. 3.2).

A screenshot of a code editor with multiple tabs. The active tab is 'mc [d... x' and it shows the contents of '/home/domingosjoseramo-h-pc/lab06/lab6-1.asm'. The code is assembly language. The editor has a dark theme and a status bar at the bottom right showing '173/173' and '100%'.

```
mc [d... x mc [d... x mc [d... x mc [d... x domin... x
/home/domingosjoseramo-h-pc/lab06/lab6-1.asm 173/173 100%
%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

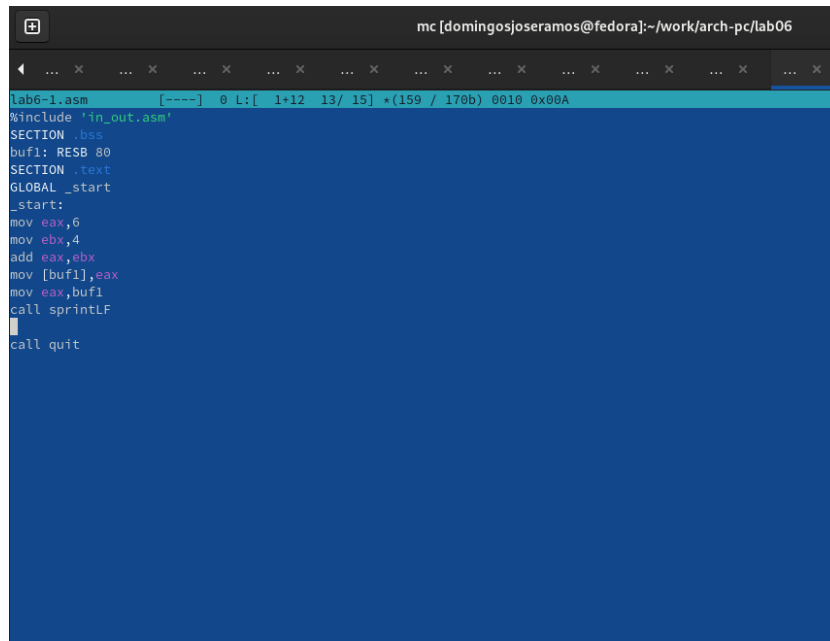
Создаем исполняемый файл и запускаем его (рис. fig. 3.3).

A screenshot of a terminal window showing the compilation and execution of the assembly file. The user is 'domingosjoseramos' on a 'fedora' machine. The commands executed are 'nasm -f elf lab6-1.asm', 'ld -m elf_i386 -o lab6-1 lab6-1.o', and './lab6-1'. The output shows the file was compiled and executed successfully.

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

Рис. 3.3: Запускаем файл и смотрим на его работу

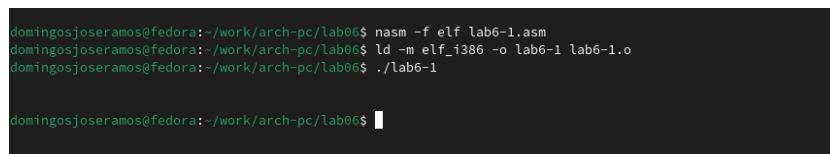
Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис. fig. 3.4).

A screenshot of a text editor window titled 'mc [domingosjoseramos@fedora]:~/work/arch-pc/lab06'. The editor displays assembly code for 'lab6-1.asm'. The code includes a comment line '[---] 0 L:[1+12 13/ 15] *(159 / 170b) 0010 0x00A', followed by '%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', and 'call quit'.

```
lab6-1.asm [---] 0 L:[ 1+12 13/ 15] *(159 / 170b) 0010 0x00A
%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. fig. 3.5).

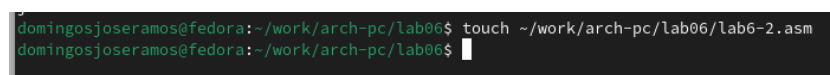
A screenshot of a terminal window showing the compilation and execution of 'lab6-1.asm'. The commands and their outputs are: 'nasm -f elf lab6-1.asm', 'ld -m elf_i386 -o lab6-1 lab6-1.o', and './lab6-1'.

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

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

Рис. 3.5: Запускаем файл и смотрим на его работу

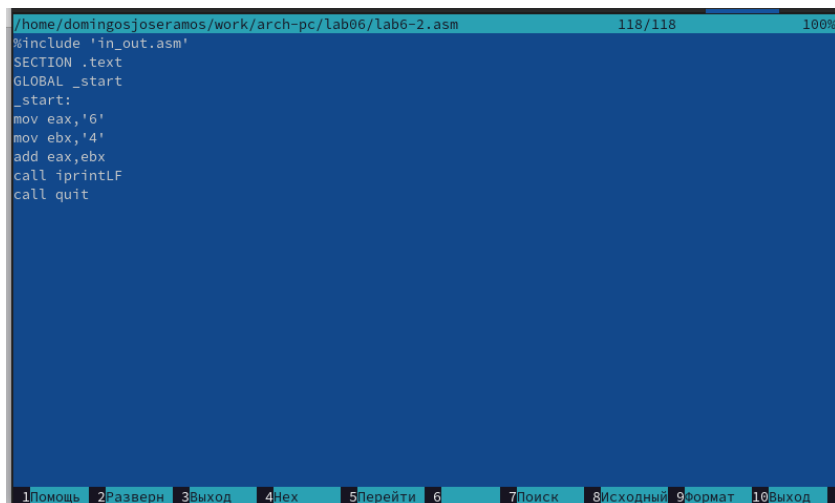
Создаем новый файл в каталоге (рис. fig. 3.6).

A screenshot of a terminal window showing the creation of a new file 'lab6-2.asm' using the 'touch' command. The command is 'touch ~/work/arch-pc/lab06/lab6-2.asm'.

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

Рис. 3.6: Создаем файл

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



```
/home/domingosjoseramos/work/arch-pc/lab06/lab6-2.asm 118/118 100%
#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

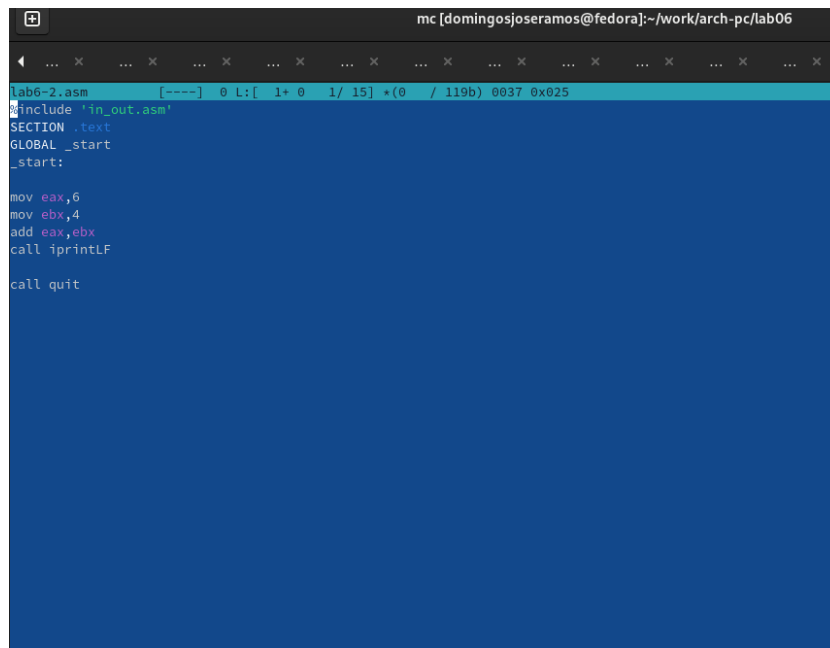
Создаем исполняемый файл и запускаем его (рис. fig. 3.8).



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

Рис. 3.8: Смотрим на работу программы

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

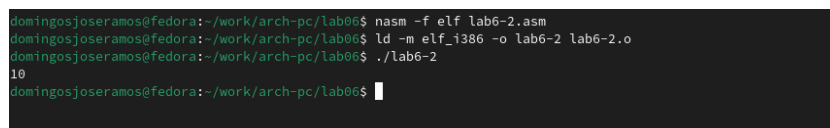
A screenshot of a text editor window titled 'mc [domingosjoseramos@fedora]:~/work/arch-pc/lab06'. The editor shows the file 'lab6-2.asm' with the following content:

```
lab6-2.asm [----] 0 L: [ 1+ 0 1/ 15] +(0 / 119b) 0037 0x025
#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

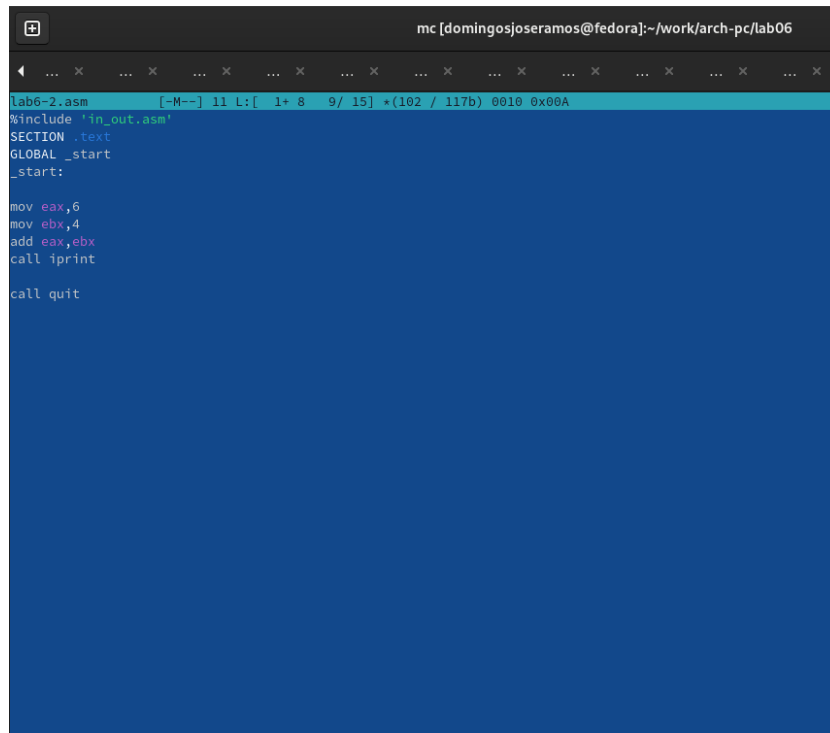
Создаем исполняемый файл и запускаем его (рис. fig. 3.10).

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

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

Рис. 3.10: Смотрим на работу программы

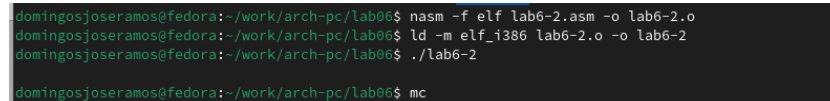
Снова открываем файл для редактирования и меняем `iprintLF` на `iprint` (рис. fig. 3.11).

A screenshot of a code editor window titled 'mc [domingosjoseramos@fedora]:~/work/arch-pc/lab06'. The editor shows the contents of 'lab6-2.asm'. The code includes a comment line, an include directive, a section directive, a global directive, and several assembly instructions. The line numbers 11 through 15 are visible on the left margin.

```
lab6-2.asm      [-M--] 11 L:[ 1+ 8  9/ 15] *(102 / 117b) 0010 0x00A
#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. fig. 3.12).

A screenshot of a terminal window showing the steps to compile and run the assembly program. The user runs 'nasm' to create an object file, 'ld' to create an executable, and 'mc' to run it.

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

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

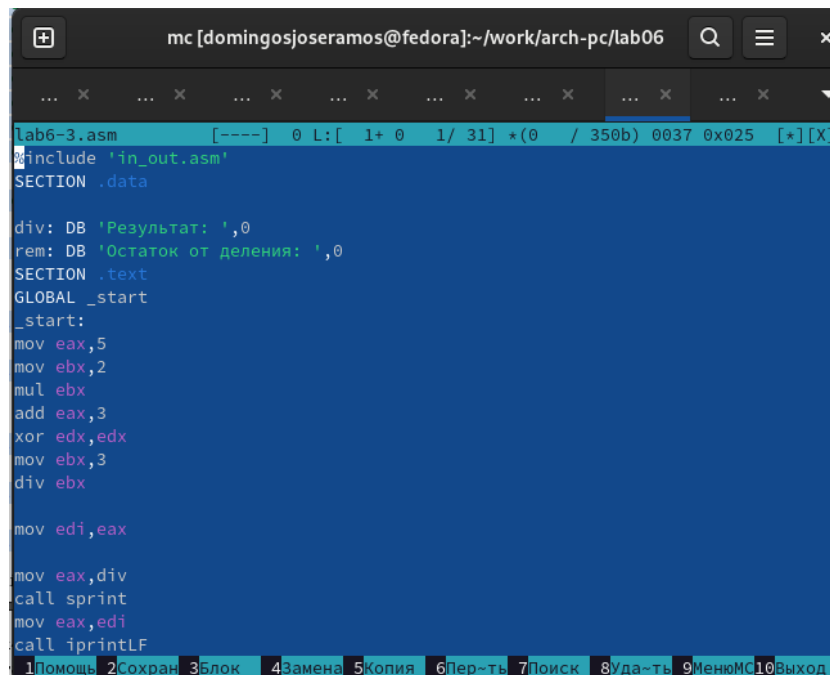
Создаем новый файл в каталоге (рис. fig. 3.13).

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

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

Рис. 3.13: Создаем файл

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



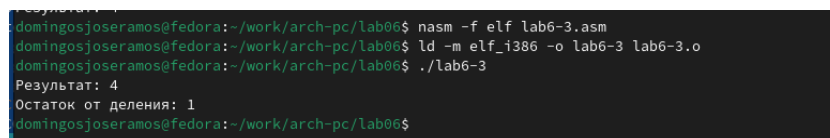
```
lab6-3.asm [----] 0 L: [ 1+ 0 1/ 31] *(0 / 350b) 0037 0x025 [*][X]
%include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,5
mov ebx,2
mul ebx
add eax,3
xor ecx,ecx
mov ebx,3
div ebx

mov edi,eax

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

Рис. 3.14: Заполняем файл

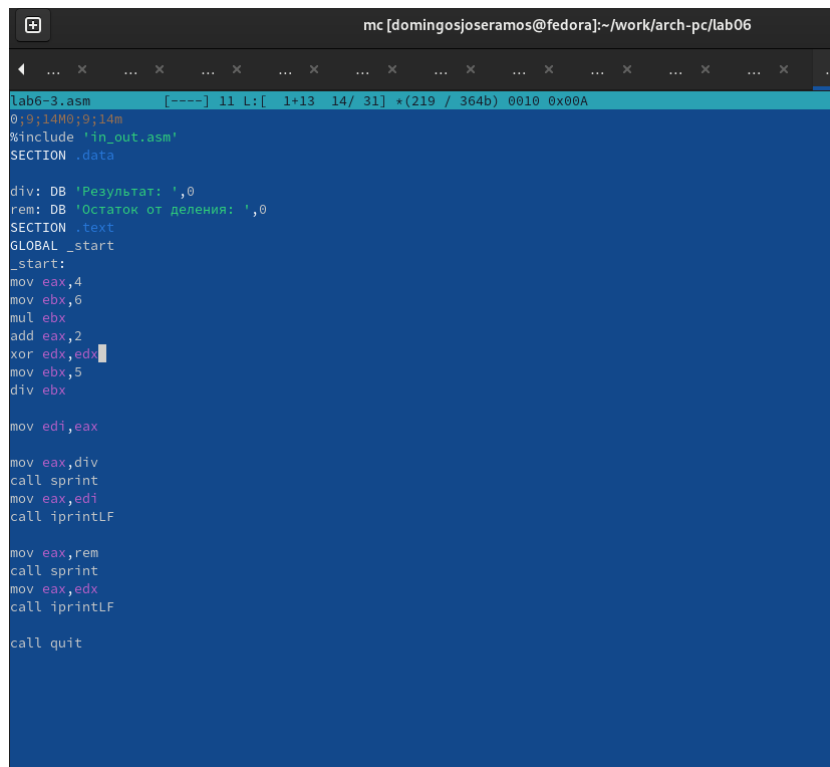
Создаем исполняемый файл и запускаем его (рис. fig. 3.15).



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

Рис. 3.15: Смотрим на результат работы программы

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



```
mc [domingosjoseramos@fedora]:~/work/arch-pc/lab06
lab6-3.asm [----] 11 L: [ 1+13 14/ 31] *(219 / 364b) 0010 0x00A
0;9;14M0;9;14m
#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

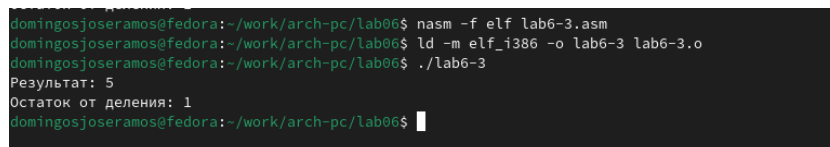
mov eax,div
call sprint
mov eax,edi
call iprintLF

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

call quit
```

Рис. 3.16: Редактируем файл

Компилируем файл и запускаем программу (рис. fig. 3.17).



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

Рис. 3.17: Смотрим на результат работы программы

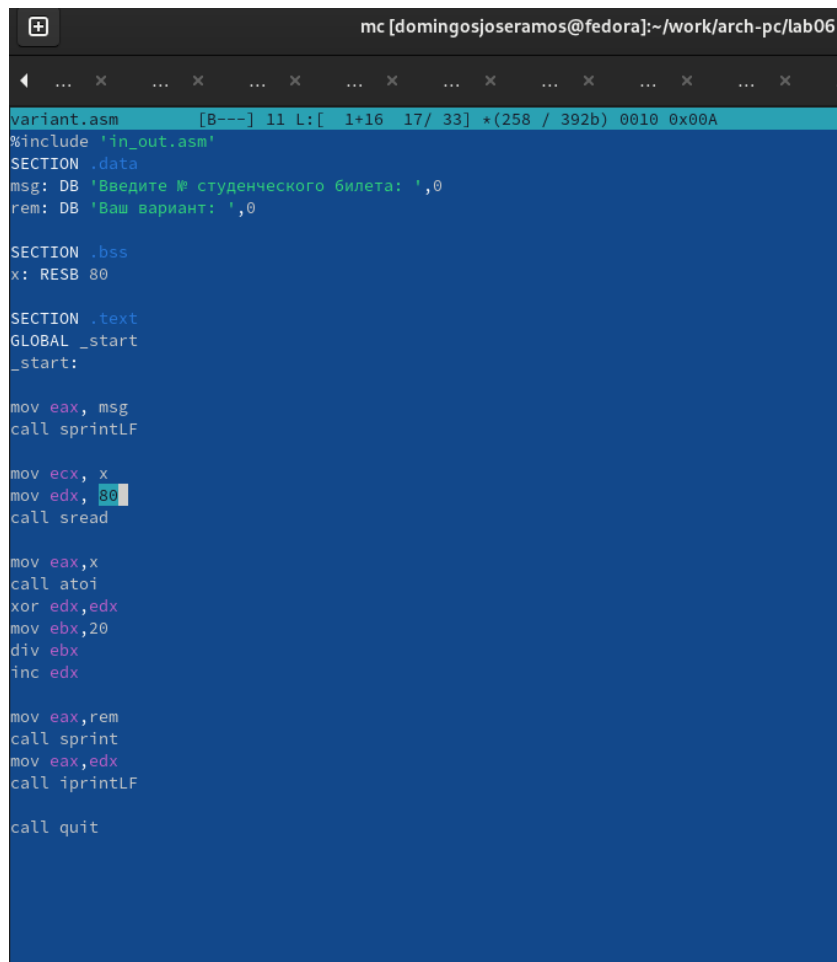
Создаем новый файл в каталоге (рис. fig. 3.18).



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

Рис. 3.18: Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.4 (рис. fig. 3.19).



```
mc [domingosjoseramos@fedora]:~/work/arch-pc/lab06
variant.asm [B---] 11 L: [ 1+16 17/ 33] *(258 / 392b) 0010 0x00A
#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 sprintLF

mov ecx, x
mov edx, 80
call sread

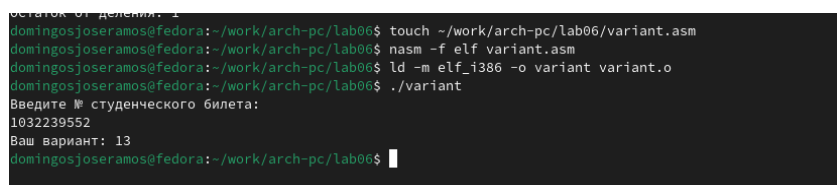
mov eax, x
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx

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

call quit
```

Рис. 3.19: Заполняем файл

Компилируем файл и запускаем его (рис. fig. 3.20).



```
domingosjoseramos@fedora: ~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
domingosjoseramos@fedora: ~/work/arch-pc/lab06$ nasm -f elf variant.asm
domingosjoseramos@fedora: ~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
domingosjoseramos@fedora: ~/work/arch-pc/lab06$ ./variant
Остаток от деления: 1
Введите № студенческого билета:
1032239552
Ваш вариант: 13
domingosjoseramos@fedora: ~/work/arch-pc/lab06$
```

Рис. 3.20: Проверяем результат работы программы

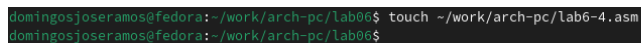
3.3 Ответы на вопросы по программе

1. Строка “mov eax,rem” и строка “call sprint” 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 есх, 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 еах и возвращает полученное число в регистре еах.
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 еах на значение регистра ebx с сохранением частного в регистре еах 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 еах,edx” передает значение остатка от деления в регистр еах. Строка “call iprintLF” вызывает процедуру iprintLF для вывода значения на экран вместе с переводом строки.

3.4 Задание для самостоятельной работы

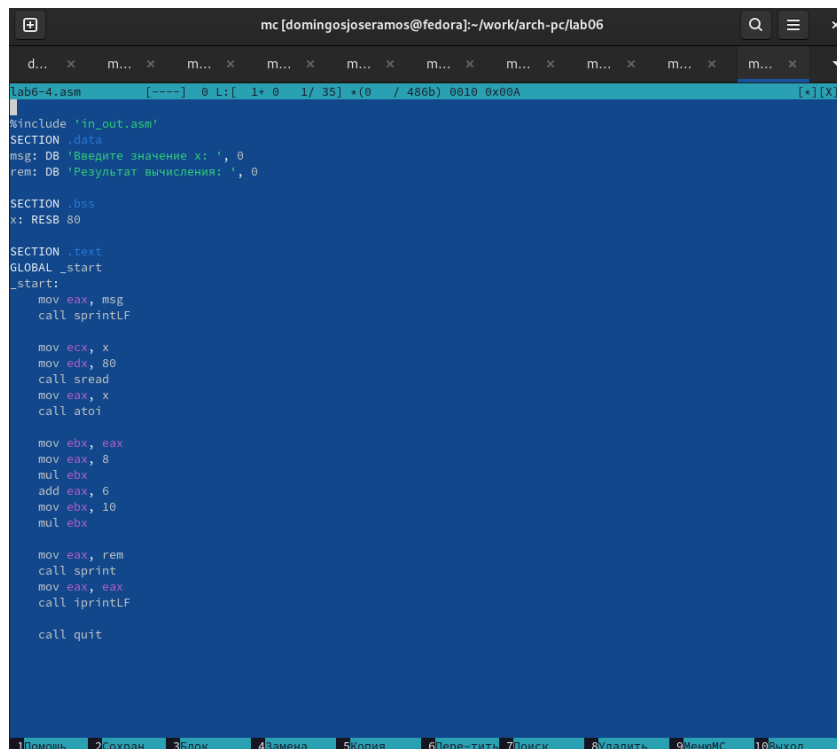
Создаем новый файл в каталоге (рис. fig. 3.21).



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

Рис. 3.21: Создаем файл

Открываем его и заполняем, чтобы решалось выражение $f(x)=(8x+6)10$ (рис. fig. 3.22).



```
mc [domingosjoseramos@fedora]:~/work/arch-pc/lab06
lab6-4.asm 0 L: 1+ 0 1/ 35] +(0 / 486b) 0010 0x00A [*)(X]
$include 'in_out.asm'
SECTION .data
msg: DB 'Введите значение x: ', 0
rem: DB 'Результат вычисления: ', 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

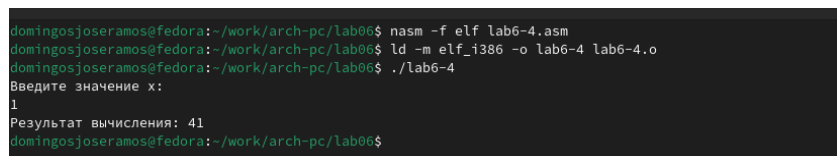
    mov ebx, eax
    mov eax, 8
    mul ebx
    add eax, 6
    mov ebx, 10
    mul ebx

    mov eax, rem
    call sprintf
    mov eax, eax
    call iprintf

    call quit
```

Рис. 3.22: Заполняем файл

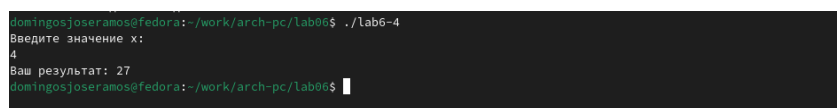
Компилируем программу и проверяем для $x=1$ (рис. fig. 3.23).



```
domingosjoseramos@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
domingosjoseramos@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
domingosjoseramos@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите значение x:
1
Результат вычисления: 41
domingosjoseramos@fedora:~/work/arch-pc/lab06$
```

Рис. 3.23: Проверяем работу программы

Компилируем программу и проверяем для $x=4$ (рис. fig. 3.24).



```
domingosjoseramos@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите значение x:
4
Ваш результат: 27
domingosjoseramos@fedora:~/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.