

Отчёта по лабораторной работе №6

Арифметические операции в NASM.

Камбунду Паулине

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.

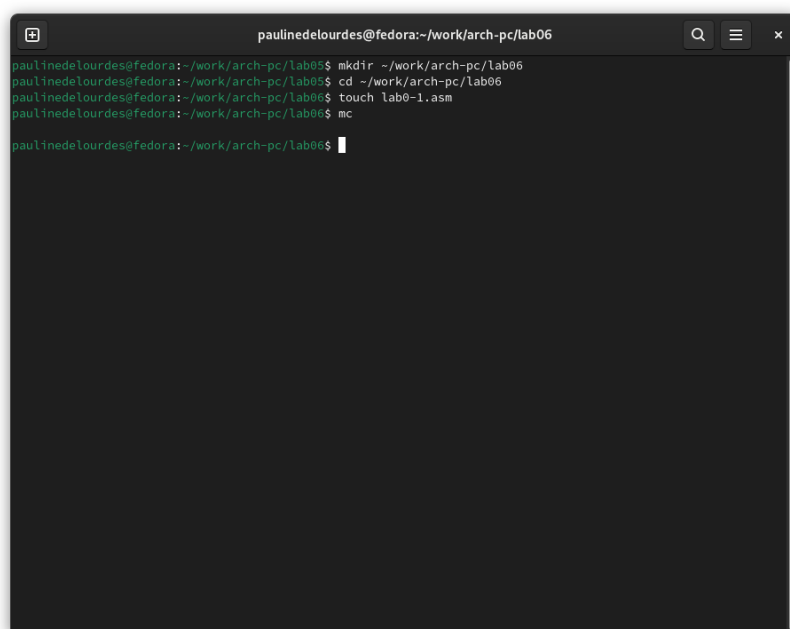
2 Задание

Написать программы для решения выражений.

3 Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

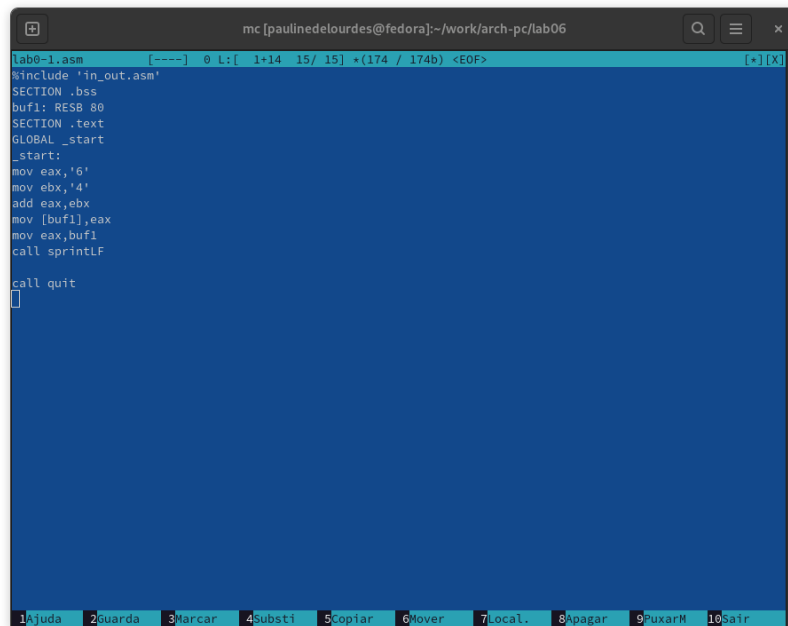
Создаем каталог для программ ЛБ6, и в нем создаем файл (рис. fig. 3.1).

A terminal window with a dark background and light green text. The window title is 'paulinedelourdes@fedora:~/work/arch-pc/lab06'. The terminal shows a sequence of commands: 'mkdir ~/work/arch-pc/lab06', 'cd ~/work/arch-pc/lab06', 'touch lab0-1.asm', and 'mc'. The prompt is 'paulinedelourdes@fedora:~/work/arch-pc/lab06\$' followed by a cursor.

```
paulinedelourdes@fedora:~/work/arch-pc/lab06$ mkdir ~/work/arch-pc/lab06
paulinedelourdes@fedora:~/work/arch-pc/lab06$ cd ~/work/arch-pc/lab06
paulinedelourdes@fedora:~/work/arch-pc/lab06$ touch lab0-1.asm
paulinedelourdes@fedora:~/work/arch-pc/lab06$ mc
paulinedelourdes@fedora:~/work/arch-pc/lab06$
```

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

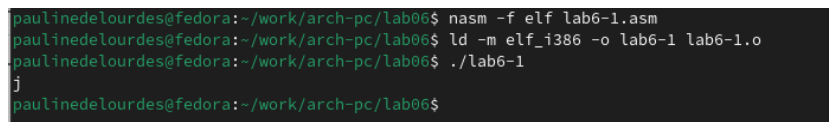
Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (рис. fig. 3.2).

A screenshot of a text editor window titled 'mc [paulinedelourdes@fedora]:~/work/arch-pc/lab06'. The editor shows the contents of 'lab6-1.asm'. The code includes a directive to include 'in_out.asm', followed by section declarations for .bss (buffer) and .text (code). It then defines a global symbol '_start' and writes assembly instructions to move values into registers, calculate an address, and call 'sprintf' to format a string. Finally, it calls 'quit'. The bottom of the window features a toolbar with icons for various editing actions like 'Ajuda', 'Guardar', 'Arcar', etc.

```
lab6-1.asm [----] 0 L: [ 1+14 15/ 15] *(174 / 174b) <EOF> [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
```

Рис. 3.2: Заполняем файл

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

A screenshot of a terminal window showing the commands used to compile and run the assembly file. The user is in the directory ~/work/arch-pc/lab06. They use 'nasm' to compile the assembly file into an object file, then 'ld' to link it into an executable, and finally run the executable with './lab6-1'.

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

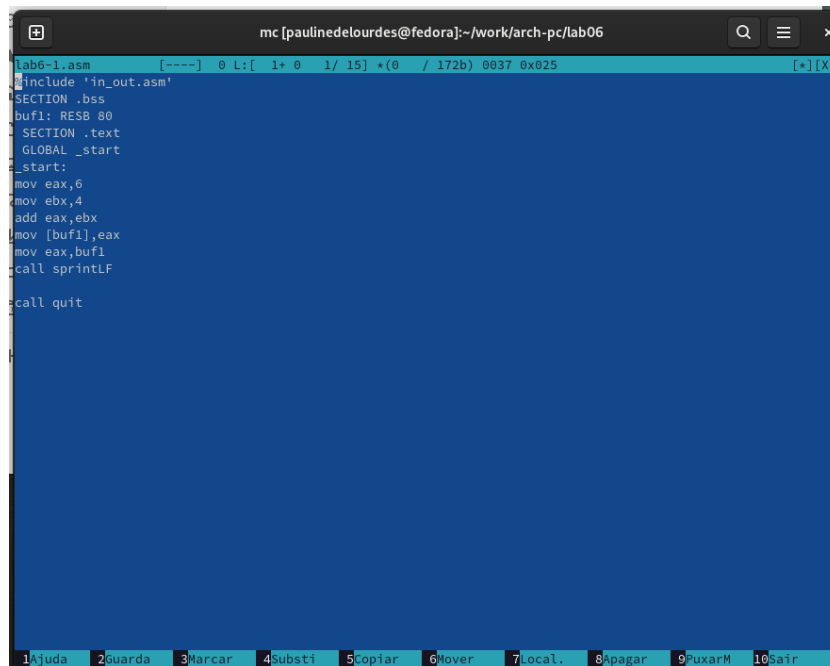


Рис. 3.4: Изменяем файл

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

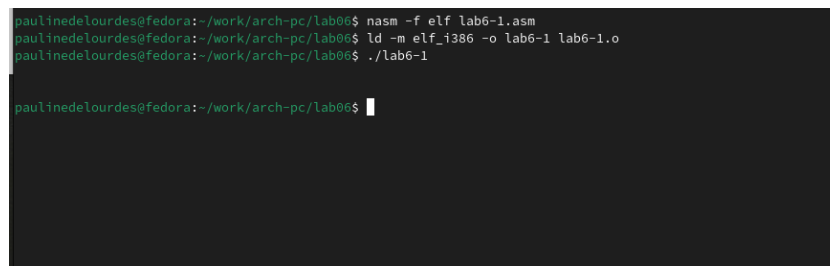


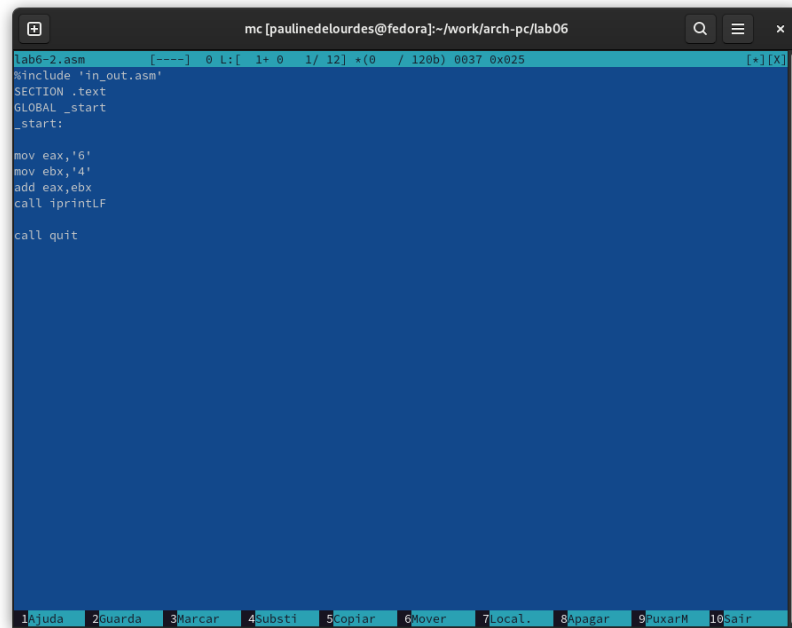
Рис. 3.5: Запускаем файл и смотрим на его работу

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



Рис. 3.6: Создаем файл

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



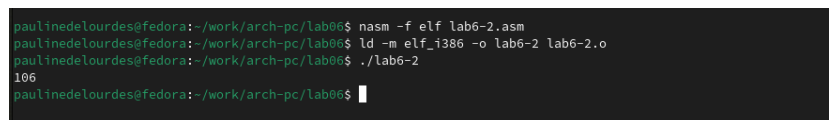
```
lab6-2.asm [----] 0 L: [ 1+ 0 1/ 12] *(0 / 120b) 0037 0x025 [+] [X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

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).



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

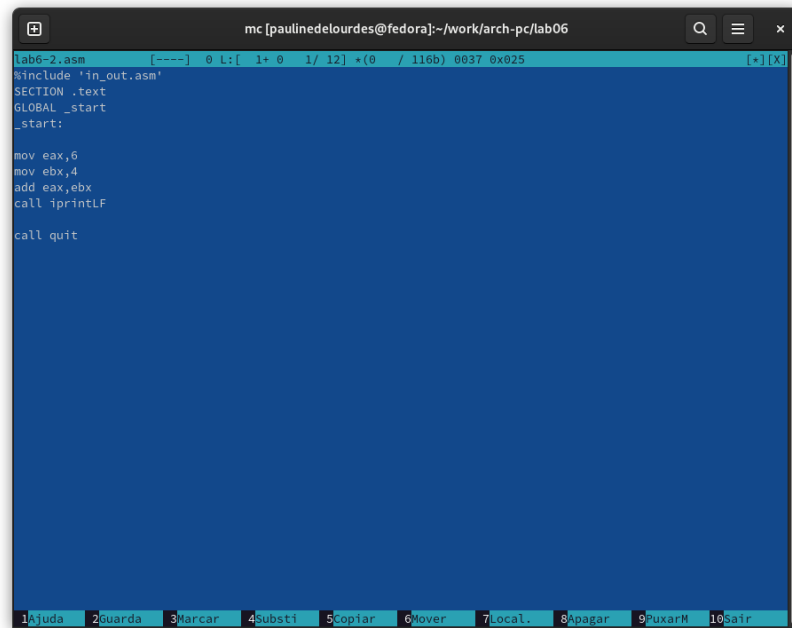


Рис. 3.9: Изменяем файл

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

```

100
paulinedelourdes@fedora:~/work/arch-pc/lab06$ mc

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

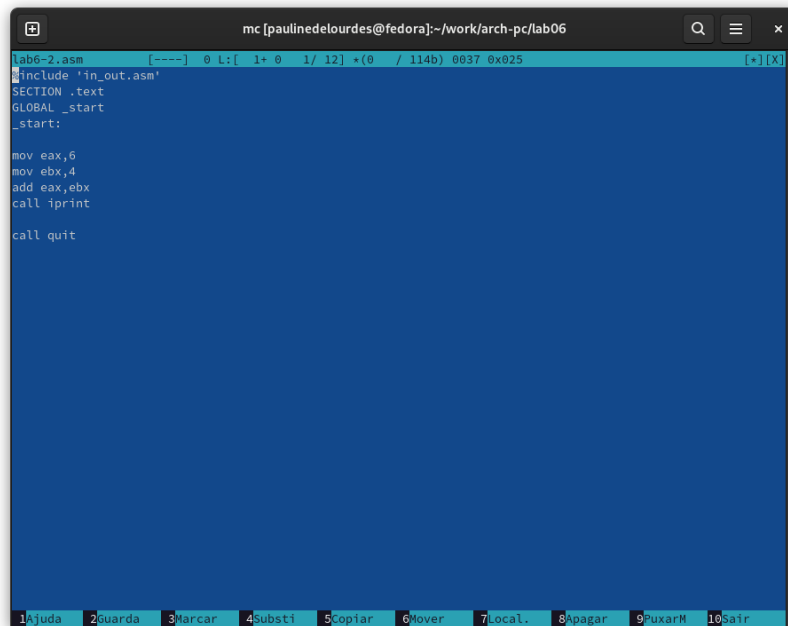


Рис. 3.11: Изменяем файл

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

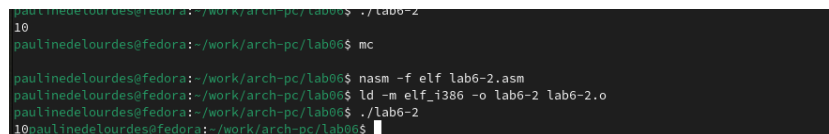


Рис. 3.12: Смотрим на работу программы

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

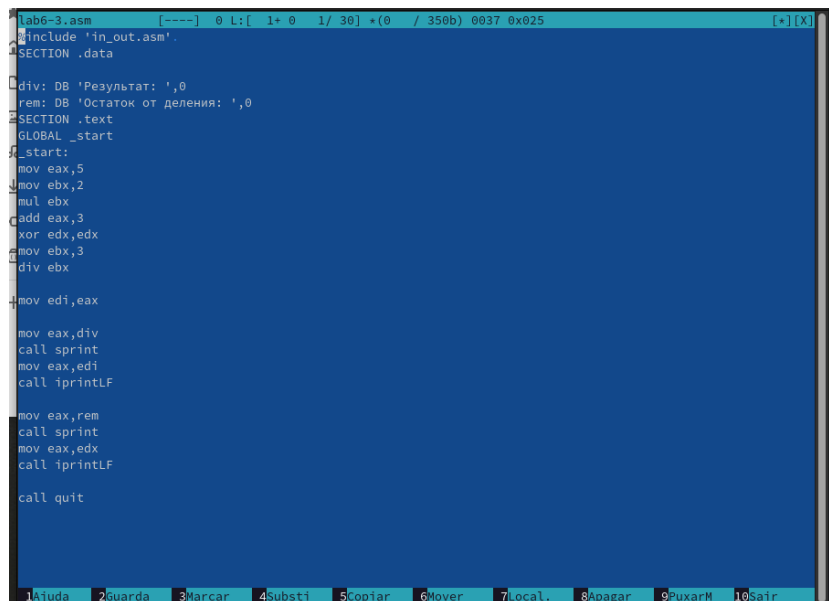
3.2 Выполнение арифметических операций в NASM

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



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/ 30] *(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 edx,edx
mov ebx,3
div ebx

mov edi,eax

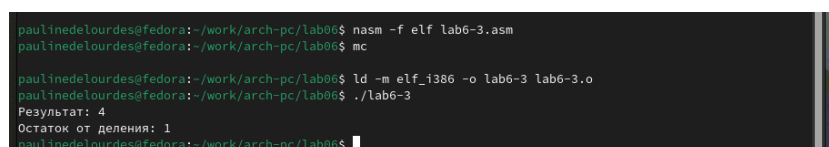
mov eax,div
call sprint
mov eax,edi
call iprintLF

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

call quit
```

Рис. 3.14: Заполняем файл

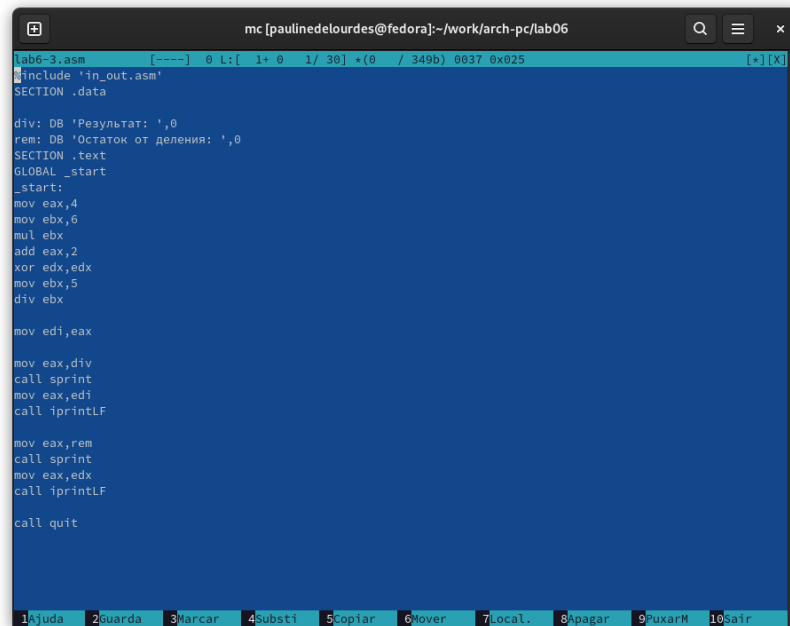
Создаем исполняемый файл и запускаем его (рис. fig. 3.15).



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



```
lab6-3.asm 0 L: [ 1+ 0 1/ 30] *(0 / 349b) 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,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

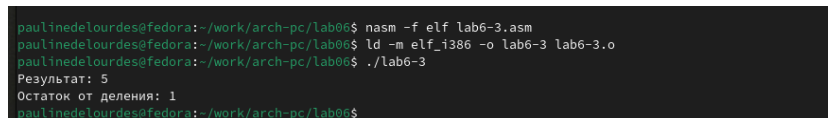
mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
```

1Ajuda 2Guarda 3Arçar 4Substi 5Copiar 6Mover 7Local. 8Apagar 9PuxarM 10Sair

Рис. 3.16: Редактируем файл

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



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

Рис. 3.17: Смотрим на результат работы программы

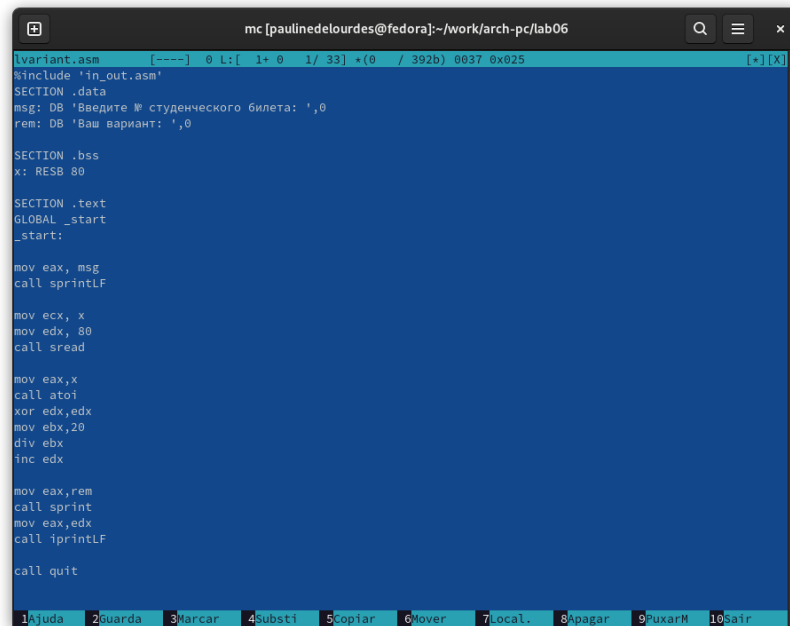
Создаем новый файл в каталоге (рис. fig. 3.18).



```
paulinedelourdes@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lvariant.asm
paulinedelourdes@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 [paulinedelourdes@fedora:~/work/arch-pc/lab06]
[variant.asm] [-----] 0 L: [ 1+ 0 1/ 33] *(0 / 392b) 0037 0x025 [*)(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

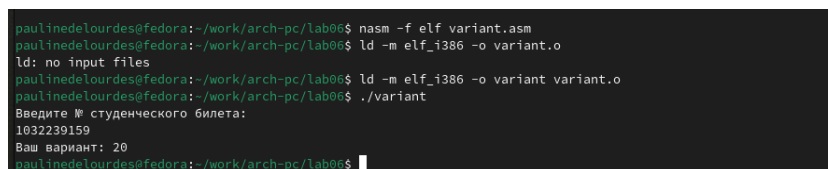
mov eax, x
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx

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

call quit
```

Рис. 3.19: Заполняем файл

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



```
paulinedelourdes@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
paulinedelourdes@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant.o
ld: no input files
paulinedelourdes@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
paulinedelourdes@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032239159
Ваш вариант: 20
paulinedelourdes@fedora:~/work/arch-pc/lab06$
```

Рис. 3.20: Проверяем результат работы программы

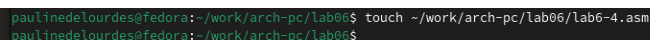
3.3 Ответы на вопросы по программе

1. Строка “mov eax,rem” и строка “call 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 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

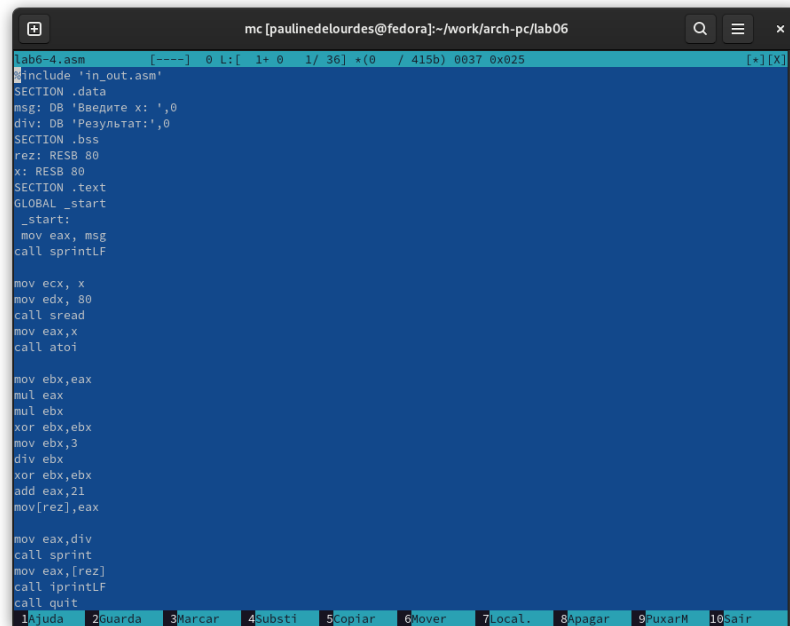
Создаем новый файл в каталоге (рис. fig. 3.21).



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

Рис. 3.21: Создаем файл

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



```
lab6-4.asm [----] 0 L: [ 1+ 0 1/ 36] *(0 / 415b) 0037 0x025 [x] [X]
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

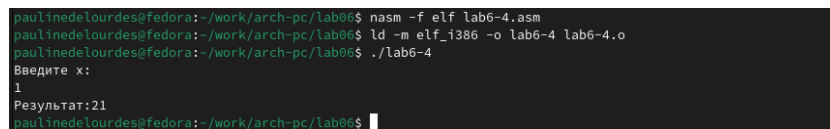
mov ecx, x
mov edx, 80
call read
mov eax, x
call atoi

mov ebx, eax
mul eax, ebx
mul ebx
xor ebx, ebx
mov ebx, 3
div ebx
xor ebx, ebx
add eax, 21
mov [rez], eax

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

Рис. 3.22: Заполняем файл

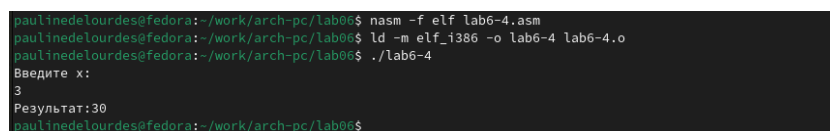
Компилируем программу и проверяем для $x=1$ (рис. fig. 3.23).



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

Рис. 3.23: Проверяем работу программы

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



```
paulinedelourdes@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
paulinedelourdes@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
paulinedelourdes@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
3
Результат:30
paulinedelourdes@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.