

Отчёта по лабораторной работе №6

Арифметические операции в NASM.

Спелов Андрей Николаевич

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	18

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	17

1 Цель работы

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

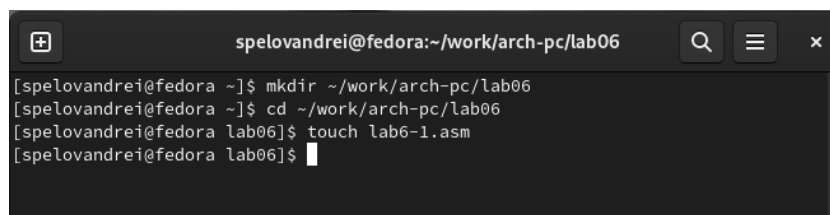
2 Задание

Написать программы для решения выражений.

3 Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

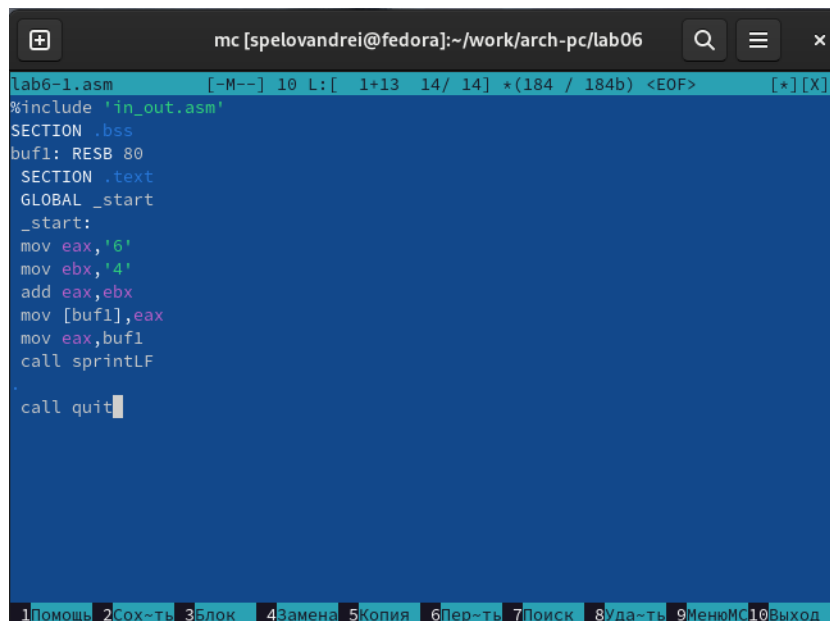
Создаем каталог для программ ЛБ6, и в нем создаем файл (рис. 3.1).

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

```
[spelovandrei@fedora ~]$ mkdir ~/work/arch-pc/lab06
[spelovandrei@fedora ~]$ cd ~/work/arch-pc/lab06
[spelovandrei@fedora lab06]$ touch lab6-1.asm
[spelovandrei@fedora lab06]$
```

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

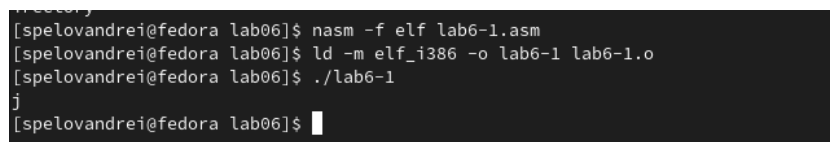
Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (рис. 3.2).



```
lab6-1.asm [-M--] 10 L: [ 1+13 14/ 14] *(184 / 184b) <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 sprintLF
call quit
```

Рис. 3.2: Заполняем файл

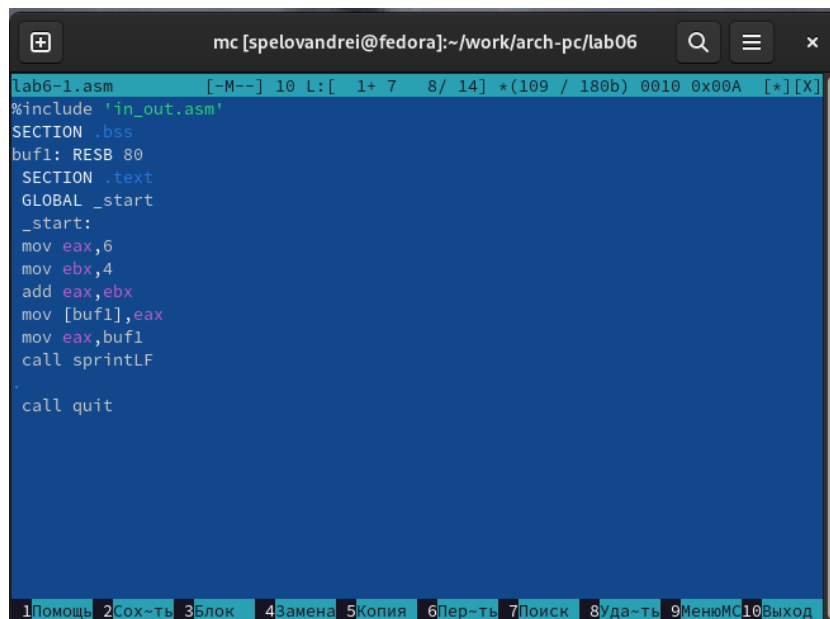
Создаем исполняемый файл и запускаем его (рис. 3.3).



```
[spelovandrei@fedora lab06]$ nasm -f elf lab6-1.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[spelovandrei@fedora lab06]$ ./lab6-1
j
[spelovandrei@fedora 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).

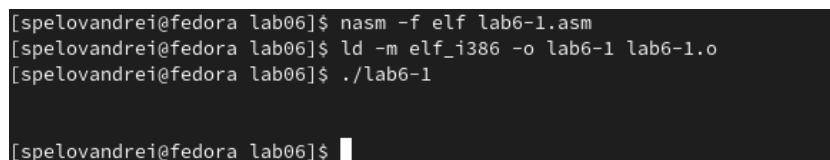
A screenshot of a nano text editor window. The title bar shows the user 'mc [spelovandrei@fedora]:~/work/arch-pc/lab06'. The editor is editing a file named 'lab6-1.asm'. The code content is as follows:

```
lab6-1.asm [-M--] 10 L:[ 1+ 7 8/ 14] *(109 / 180b) 0010 0x00A [*][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
```

At the bottom of the window, there is a menu bar with the following items: 1Помощь, 2Сох-ть, 3Блок, 4Замена, 5Копия, 6Пер-ть, 7Поиск, 8Уда-ть, 9МенюМС, 10Выход.

Рис. 3.4: Изменяем файл

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

A screenshot of a terminal window. The user is at the prompt '[spelovandrei@fedora lab06]\$'. They enter the following commands:

```
[spelovandrei@fedora lab06]$ nasm -f elf lab6-1.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[spelovandrei@fedora lab06]$ ./lab6-1
```

The terminal shows the output of these commands, and the prompt returns after the execution of './lab6-1'.

Рис. 3.5: Запускаем файл и смотрим на его работу

Создаем новый файл в каталоге (рис. 3.6).

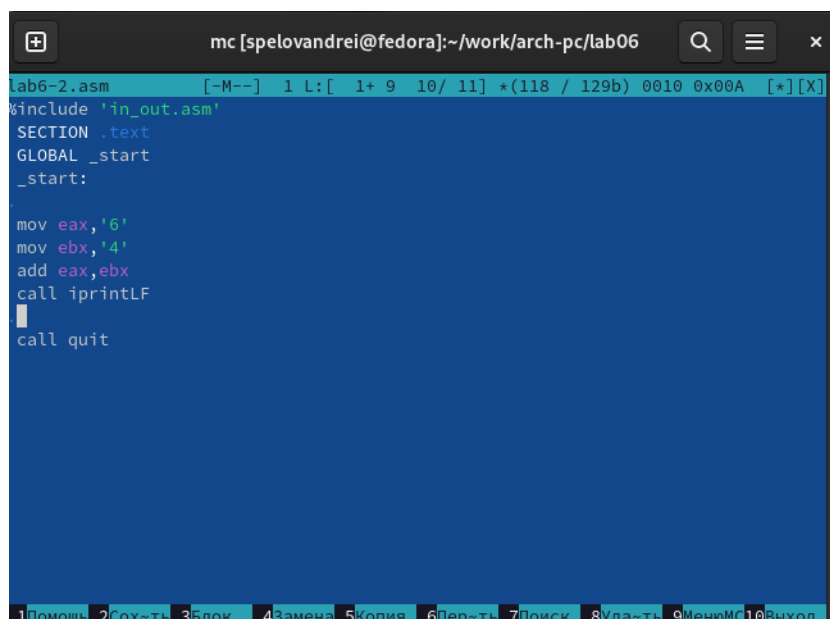
A screenshot of a terminal window. The user is at the prompt '[spelovandrei@fedora lab06]\$'. They enter the command:

```
[spelovandrei@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm
```

The terminal shows the output of this command, and the prompt returns.

Рис. 3.6: Создаем файл

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



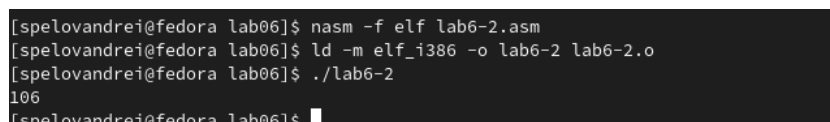
The screenshot shows a text editor window titled 'mc [spelovandrei@fedora]:~/work/arch-pc/lab06'. The editor contains the following assembly code:

```
lab6-2.asm [-M--] 1 L: [ 1+ 9 10/ 11] *(118 / 129b) 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
```

At the bottom of the window, there is a menu bar with the following items: 1Помощь, 2Сох-ть, 3Блок, 4Замена, 5Копия, 6Пер-ть, 7Поиск, 8Уда-ть, 9МенюМС10Выход.

Рис. 3.7: Заполняем файл

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

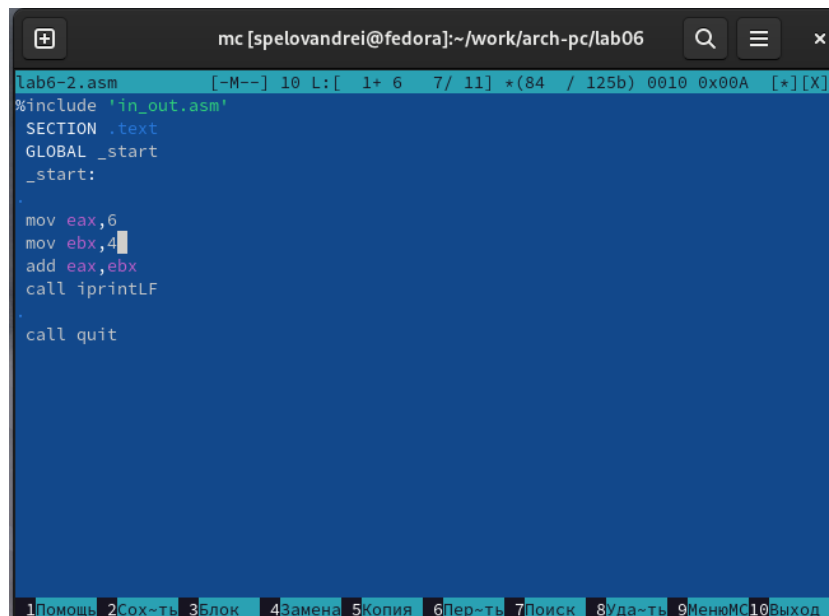


The screenshot shows a terminal window with the following commands and output:

```
[spelovandrei@fedora lab06]$ nasm -f elf lab6-2.asm  
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o  
[spelovandrei@fedora lab06]$ ./lab6-2  
106  
[spelovandrei@fedora lab06]$
```

Рис. 3.8: Смотрим на работу программы

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



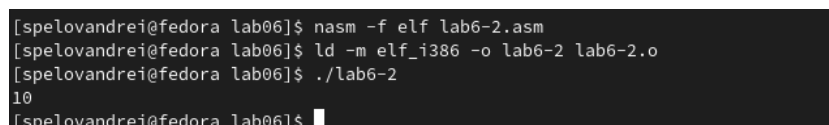
```
lab6-2.asm [-M--] 10 L: [ 1+ 6 7/ 11] *(84 / 125b) 0010 0x00A [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 3.9: Изменяем файл

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



```
[spelovandrei@fedora lab06]$ nasm -f elf lab6-2.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[spelovandrei@fedora lab06]$ ./lab6-2
10
[spelovandrei@fedora lab06]$
```

Рис. 3.10: Смотрим на работу программы

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

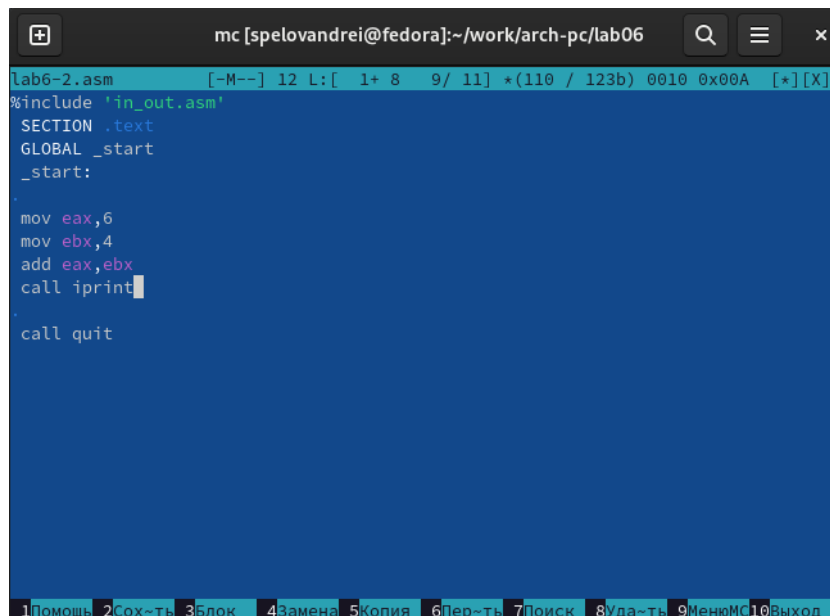


Рис. 3.11: Изменяем файл

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

```

[spelovandrei@fedora lab06]$ nasm -f elf lab6-2.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[spelovandrei@fedora lab06]$ ./lab6-2
10[spelovandrei@fedora 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

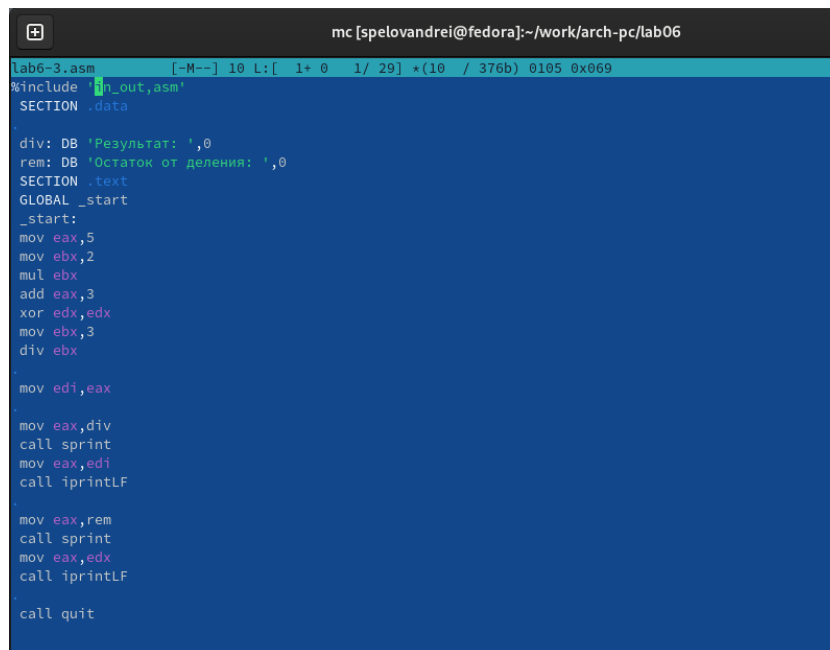
Создаем новый файл в каталоге (рис. 3.13).

```

[spelovandrei@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-3.asm
[spelovandrei@fedora lab06]$
  
```

Рис. 3.13: Создаем файл

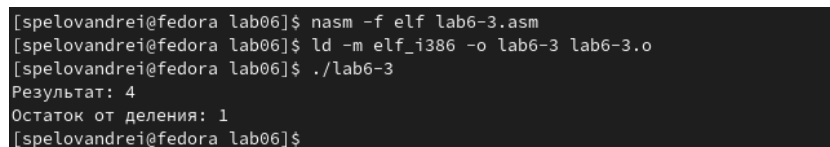
Открываем файл и редактируем в соответствии с листингом 6.3 (рис. 3.14).



```
mc [spelovandrei@fedora]:~/work/arch-pc/lab06
lab6-3.asm [-M--] 10 L: [ 1+ 0 1/ 29] *(10 / 376b) 0105 0x069
#include 'lab_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

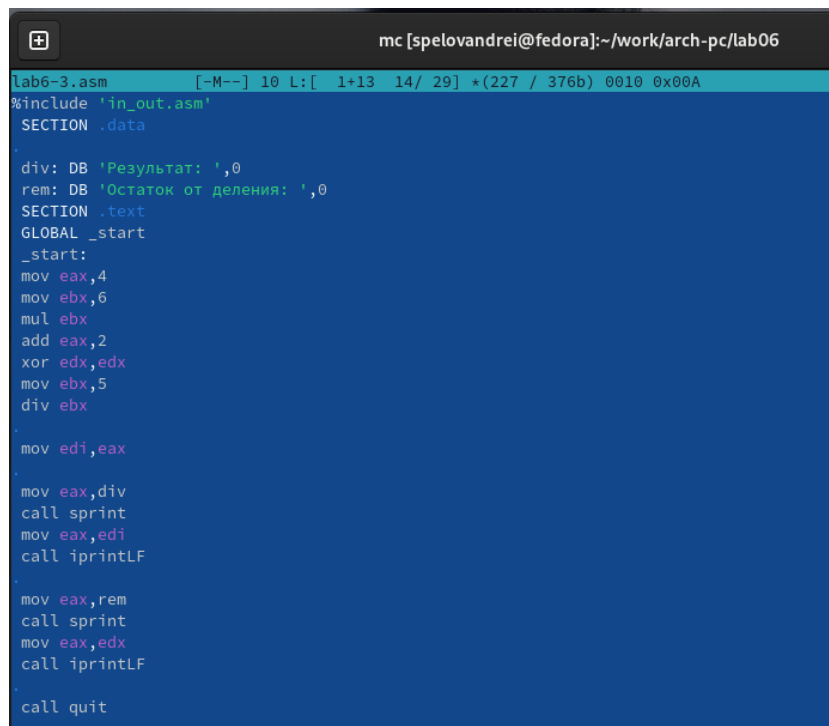
Создаем исполняемый файл и запускаем его (рис. 3.15).



```
[spelovandrei@fedora lab06]$ nasm -f elf lab6-3.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[spelovandrei@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
[spelovandrei@fedora 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. 3.16).



```
mc [spelovandrei@fedora]:~/work/arch-pc/lab06
lab6-3.asm [-M--] 10 L: [ 1+13 14/ 29] *(227 / 376b) 0010 0x00A
#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

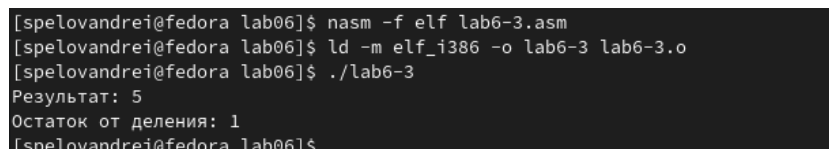
mov eax,div
call sprint
mov eax,edi
call iprintLF

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

call quit
```

Рис. 3.16: Редактируем файл

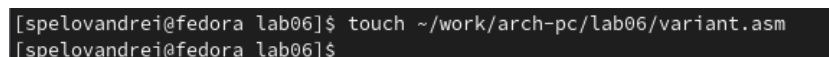
Компилируем файл и запускаем программу (рис. 3.17).



```
[spelovandrei@fedora lab06]$ nasm -f elf lab6-3.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[spelovandrei@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[spelovandrei@fedora lab06]$
```

Рис. 3.17: Смотрим на результат работы программы

Создаем новый файл в каталоге (рис. 3.18).



```
[spelovandrei@fedora lab06]$ touch ~/work/arch-pc/lab06/variant.asm
[spelovandrei@fedora lab06]$
```

Рис. 3.18: Создаем файл

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

```
mc [spelovandrei@fedora]:~/work/arch-pc/lab06
variant.asm [-M--] 10 L:[ 1+31 32/ 32] *(421 / 421b) <EOF>
%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 _outt
```

Рис. 3.19: Заполняем файл

Компилируем файл и запускаем его (рис. 3.20).

```
[spelovandrei@fedora lab06]$ nasm -f elf variant.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[spelovandrei@fedora lab06]$ ./variant
Введите № студенческого билета:
1132231839
Ваш вариант: 20
[spelovandrei@fedora 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 `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).

```
[spelovandrei@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-4.asm  
[spelovandrei@fedora lab06]$
```

Рис. 3.21: Создаем файл

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

```
mc [spelovandrei@fedora]:~/work/arch-pc/lab06
lab6-4.asm [----] 9 L: [ 1+19 20/ 34] *(290 / 468b) 0044 0x02C
#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 sprintLF
..
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,3
div ebx
xor ebx,ebx
add eax,21
mov [rez],eax
..
mov eax,div
call sprint
mov eax,[rez]
call iprintLF
call quit
```

Рис. 3.22: Заполняем файл

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

```
[spelovandrei@fedora lab06]$ nasm -f elf lab6-4.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[spelovandrei@fedora lab06]$ ./lab6-4
Введите x:
1
Результат: 21
[spelovandrei@fedora lab06]$
```

Рис. 3.23: Проверяем работу программы

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


```
[spelovandrei@fedora lab06]$ nasm -f elf lab6-4.asm
[spelovandrei@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[spelovandrei@fedora lab06]$ ./lab6-4
Введите x:
3
Результат: 30
[spelovandrei@fedora lab06]$
```

Рис. 3.24: Проверяем работу программы

4 Выводы

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