

Отчет по лабораторной работе №6

Арифметические операции в NASM

Майоров Дмитрий Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Ответы на вопросы	16
5	Задание для самостоятельной работы	17
6	Выводы	20
	Список литературы	21

Список иллюстраций

3.1	Создаем каталог и файл в нем	7
3.2	Открываем файл и заполняем его в соответствии с листингом . . .	7
3.3	Создаем исполняемый файл и запускаем его	8
3.4	Изменяем файл: убираем кавычки с числовых значений	8
3.5	Запускаем файл	8
3.6	Создаем новый файл в каталоге	8
3.7	Открываем файл и заполняем его в соответствии с листингом . . .	9
3.8	Создаем исполняемый файл и запускаем его	9
3.9	Изменяем файл: убираем кавычки с числовых значений	10
3.10	Запускаем файл	10
3.11	Изменяем файл: меняем <code>iprintLF</code> на <code>iprint</code>	11
3.12	Создаем исполняемый файл и запускаем его	11
3.13	Создаем файл	11
3.14	Открываем файл и заполняем его в соответствии с листингом . . .	12
3.15	Создаем исполняемый файл и запускаем его	12
3.16	Открываем файл и редактируем его для вычисления выражения $f(x) = (4*6+2)/5$	13
3.17	Создаем исполняемый файл и запускаем его	13
3.18	Создаем файл	14
3.19	Открываем файл и заполняем его в соответствии с листингом . . .	14
3.20	Создаем исполняемый файл и запускаем его	15
5.1	Создаем новый файл в каталоге	17
5.2	Заполняем файл	18
5.3	Запускаем программу и проверяем ее работу для $x=1$ и $x=3$	19

Список таблиц

1 Цель работы

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

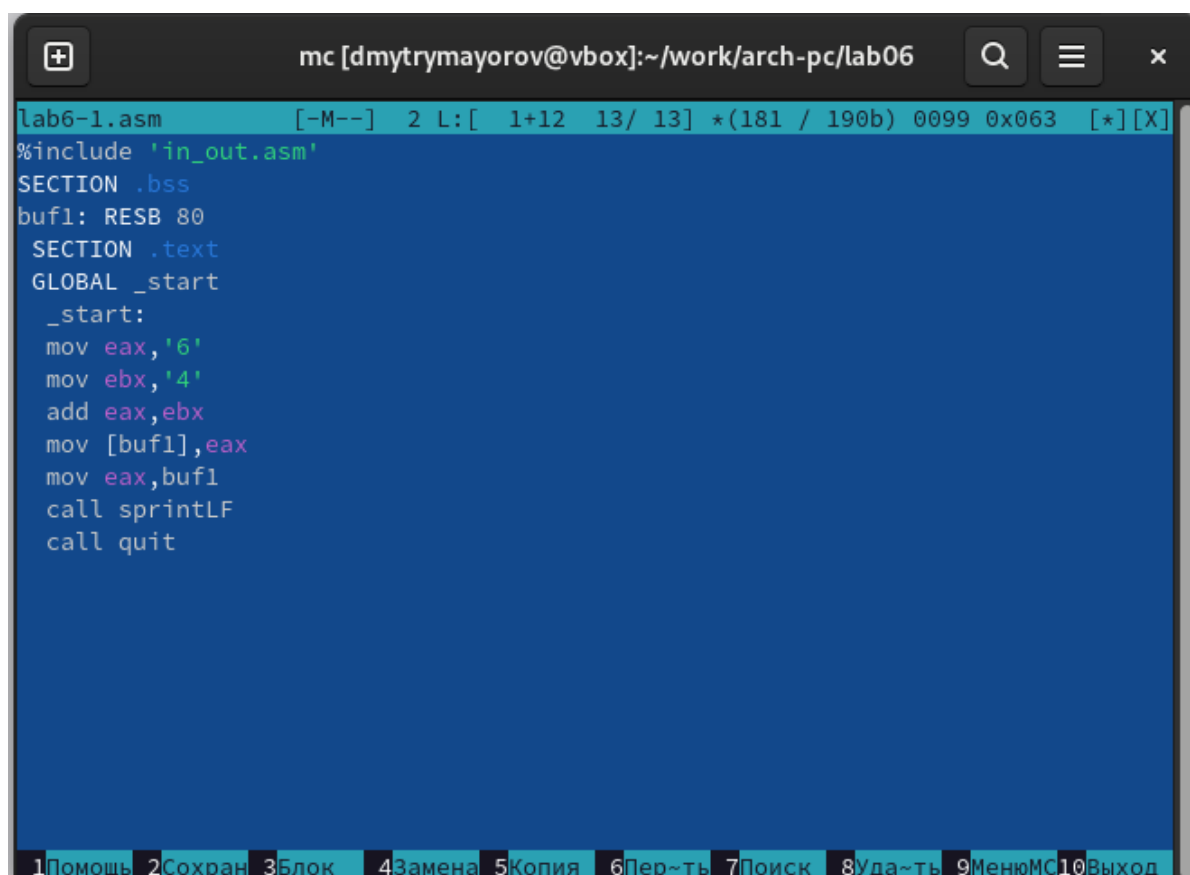
2 Задание

Написать программы для решения выражений.

3 Выполнение лабораторной работы

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

Рис. 3.1: Создаем каталог и файл в нем



The screenshot shows a text editor window titled "mc [dmytrymayorov@vbox]:~/work/arch-pc/lab06". The editor displays the assembly code for "lab6-1.asm". The code includes a header line with file encoding and line numbers, followed by assembly directives and instructions. The code is as follows:

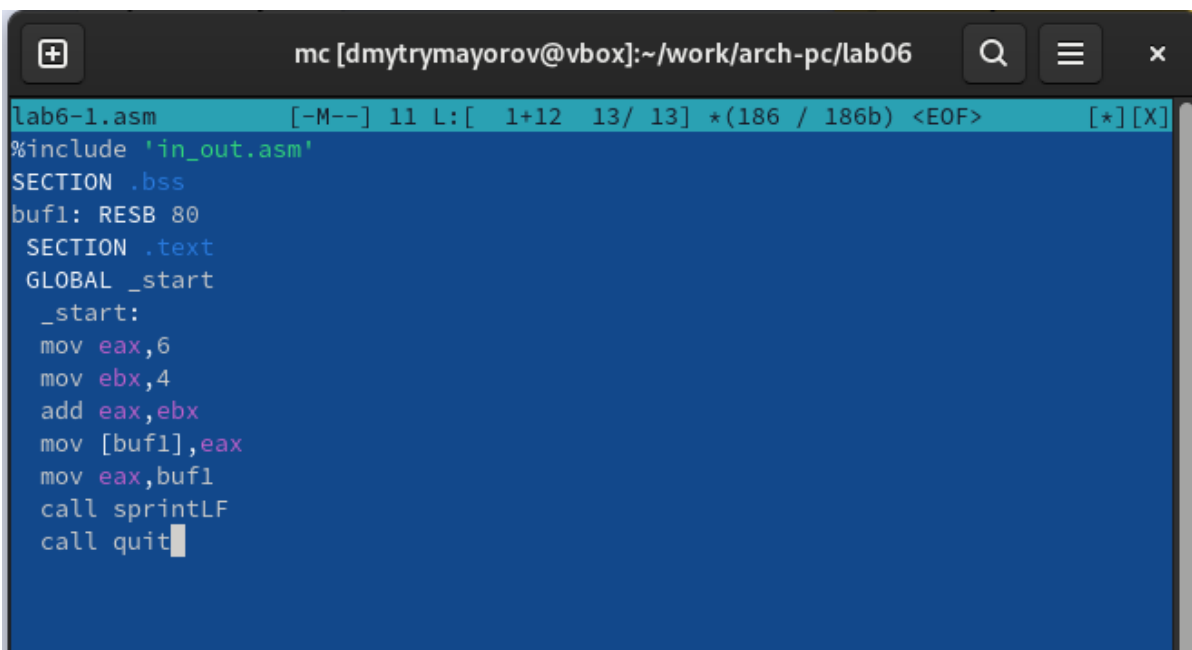
```
lab6-1.asm [-M--] 2 L:[ 1+12 13/ 13] *(181 / 190b) 0099 0x063 [*][X]
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintf
call quit
```

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

Рис. 3.2: Открываем файл и заполняем его в соответствии с листингом

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

Рис. 3.3: Создаем исполняемый файл и запускаем его



```
lab6-1.asm [-M--] 11 L:[ 1+12 13/ 13] *(186 / 186b) <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. 3.4: Изменяем файл: убираем кавычки с числовых значений

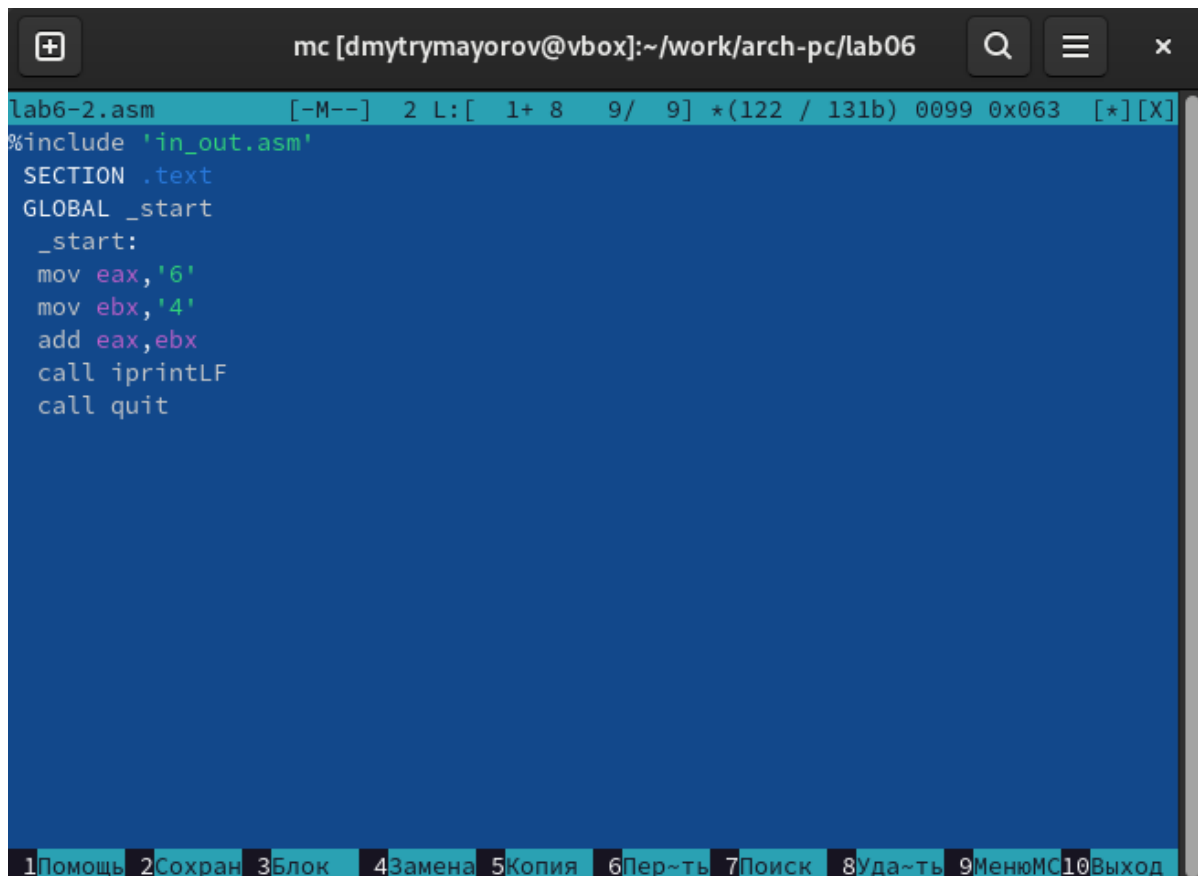
```
dmytrymayorov@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ./lab6-1

dmytrymayorov@vbox:~/work/arch-pc/lab06$
```

Рис. 3.5: Запускаем файл

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

Рис. 3.6: Создаем новый файл в каталоге

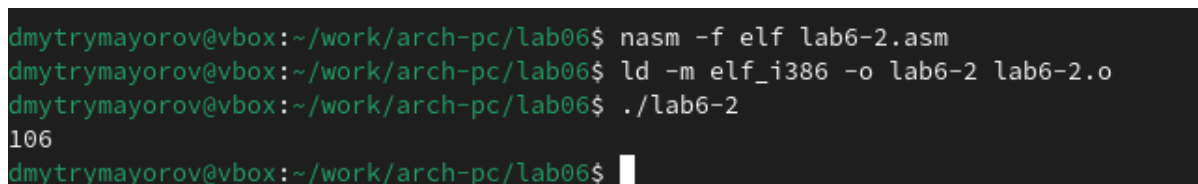


The screenshot shows a text editor window titled 'mc [dmytrymayorov@vbox]:~/work/arch-pc/lab06'. The editor is displaying the file 'lab6-2.asm'. The code content is as follows:

```
lab6-2.asm [-M--] 2 L: [ 1+ 8 9/ 9] *(122 / 131b) 0099 0x063 [*] [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, 2Сохран, 3Блок, 4Замена, 5Копия, 6Пер~ть, 7Поиск, 8Уда~ть, 9МенюМС, 10Выход.

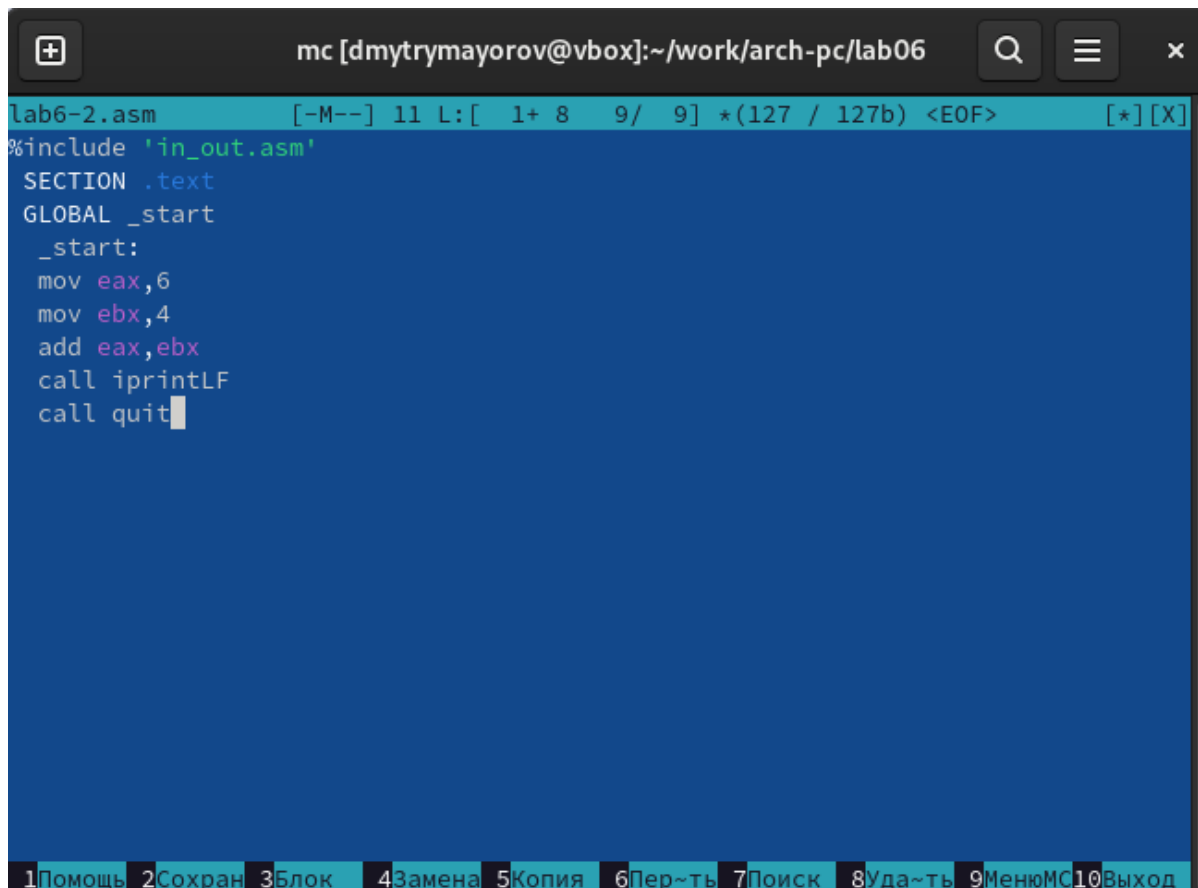
Рис. 3.7: Открываем файл и заполняем его в соответствии с листингом



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

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

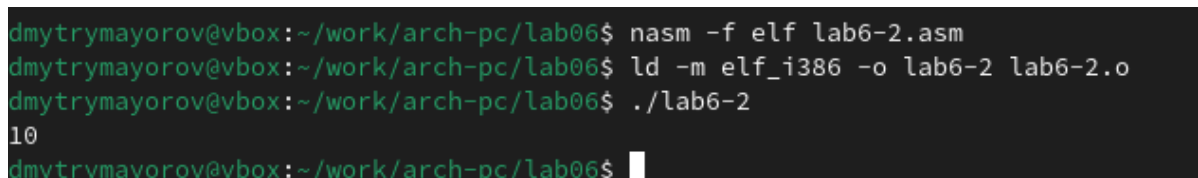
Рис. 3.8: Создаем исполняемый файл и запускаем его



```
lab6-2.asm [-M--] 11 L: [ 1+ 8 9/ 9] *(127 / 127b) <EOF> [*] [X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

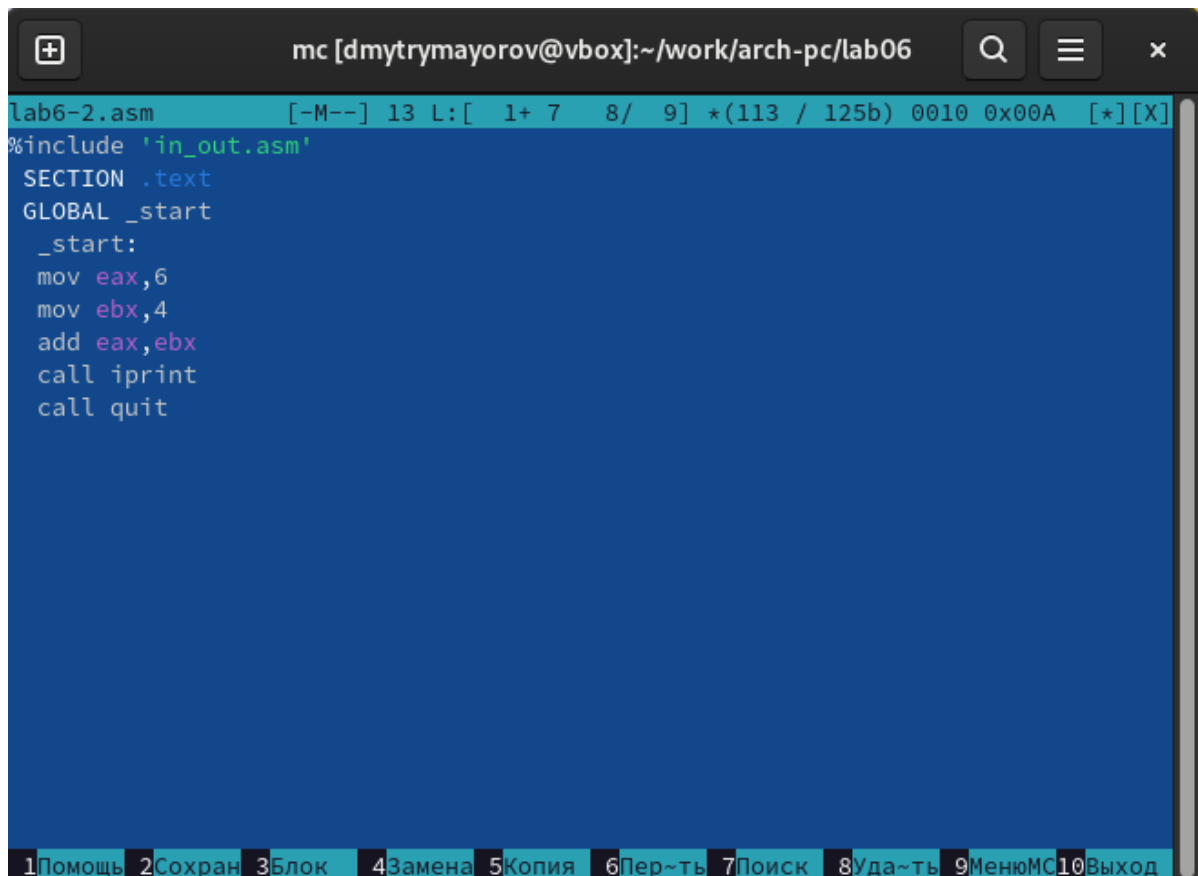
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC 10Выход

Рис. 3.9: Изменяем файл: убираем кавычки с числовых значений



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

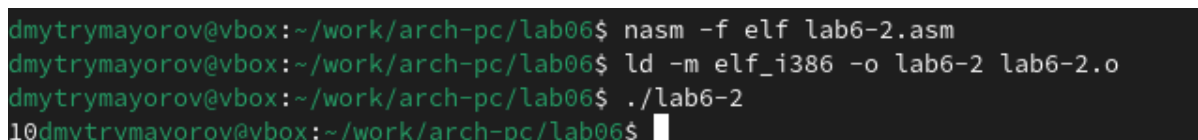
Рис. 3.10: Запускаем файл



```
lab6-2.asm [-M--] 13 L: [ 1+ 7 8/ 9] *(113 / 125b) 0010 0x00A [*] [X]
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

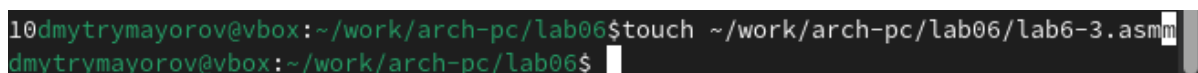
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюMC10Выход

Рис. 3.11: Изменяем файл: меняем iprintLF на iprint



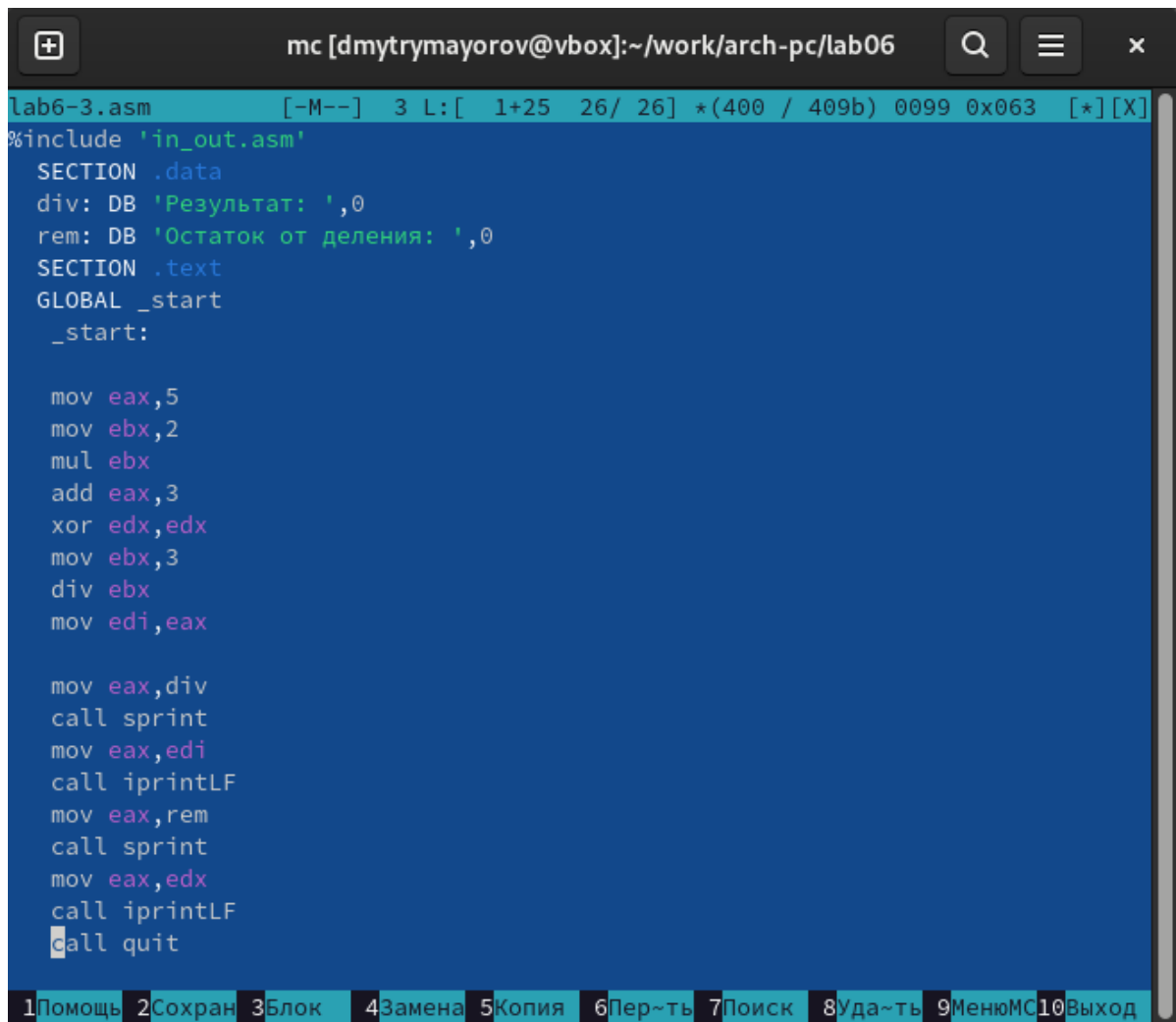
```
dmytrymayorov@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ./lab6-2
10dmytrymayorov@vbox:~/work/arch-pc/lab06$
```

Рис. 3.12: Создаем исполняемый файл и запускаем его



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

Рис. 3.13: Создаем файл



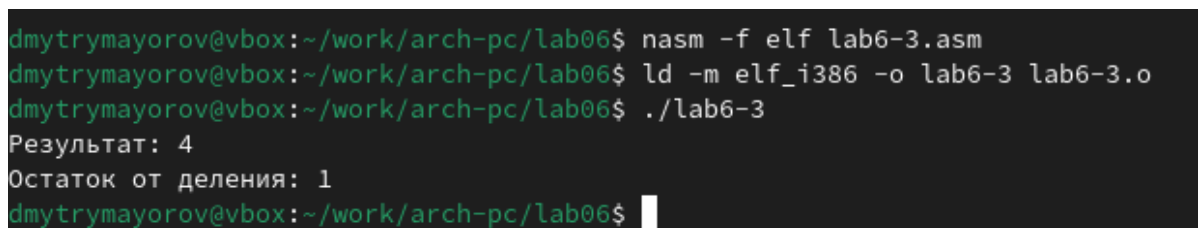
The screenshot shows a text editor window titled "mc [dmytrymayorov@vbox]:~/work/arch-pc/lab06". The editor displays the contents of "lab6-3.asm". The code includes a header section, data section with strings for "Результат:" and "Остаток от деления:", and a text section with assembly instructions. The instructions calculate 5 divided by 2, store the quotient in EDI and remainder in EDX, and then print them using "sprint" and "iprintLF" functions. A menu bar at the bottom contains options like "Помощь", "Сохран", "Блок", "Замена", "Копия", "Пер~ть", "Поиск", "Уда~ть", "МенюМС", and "Выход".

```
lab6-3.asm [-M--] 3 L: [ 1+25 26/ 26] *(400 / 409b) 0099 0x063 [*] [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
```

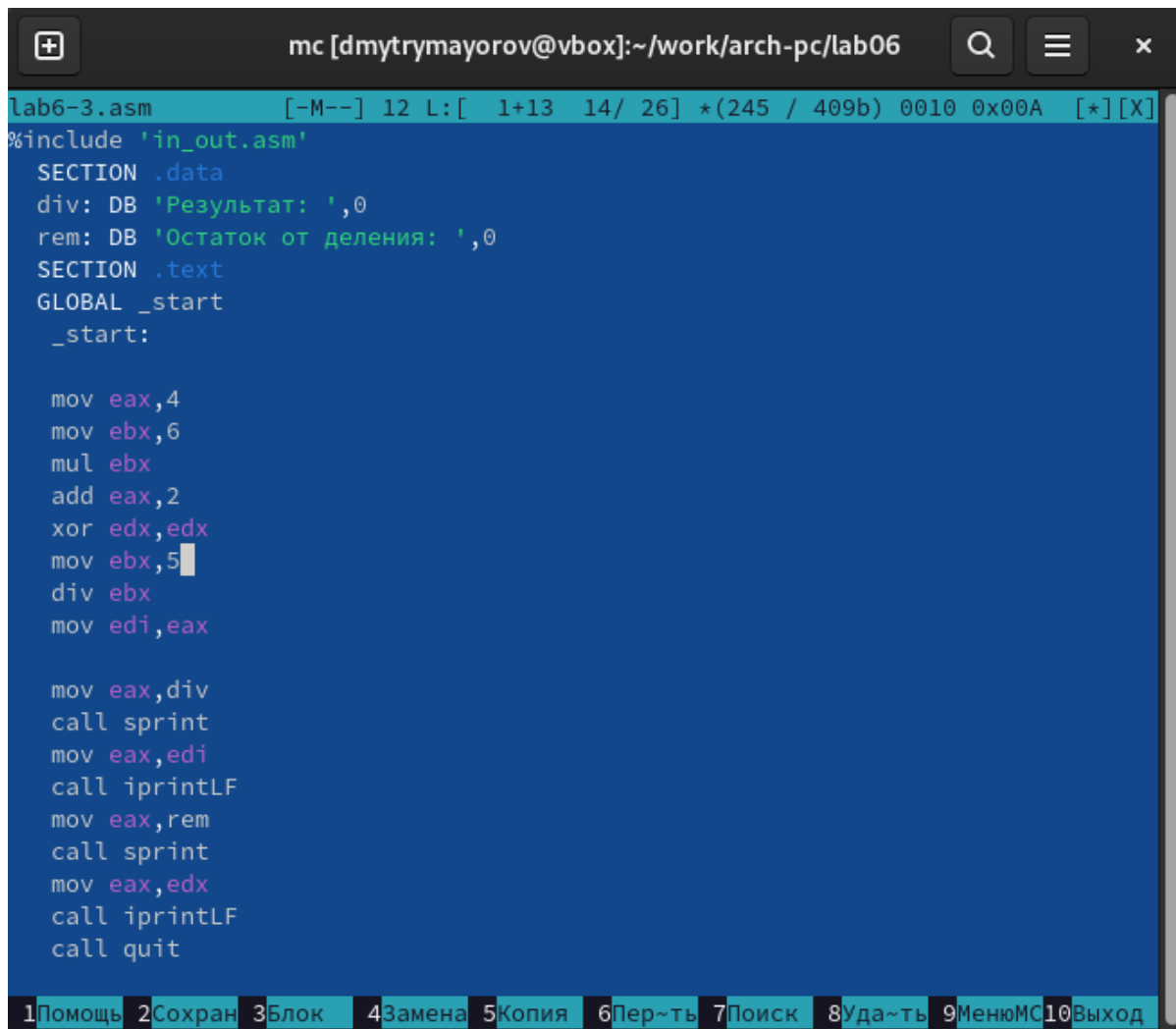
Рис. 3.14: Открываем файл и заполняем его в соответствии с листингом



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

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

Рис. 3.15: Создаем исполняемый файл и запускаем его

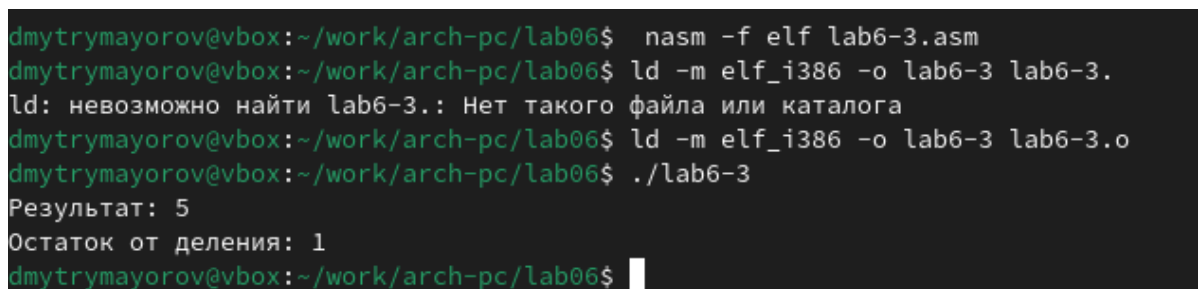


```
lab6-3.asm [-M--] 12 L: [ 1+13 14/ 26] *(245 / 409b) 0010 0x00A [*] [X]
%include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:

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
```

Рис. 3.16: Открываем файл и редактируем его для вычисления выражения $f(x) = (4 \cdot 6 + 2) / 5$



```
dmytrymayorov@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
ld: невозможно найти lab6-3.: Нет такого файла или каталога
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
dmytrymayorov@vbox:~/work/arch-pc/lab06$
```

Рис. 3.17: Создаем исполняемый файл и запускаем его

```
dmytrymayorov@vbox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
dmytrymayorov@vbox:~/work/arch-pc/lab06$
```

Рис. 3.18: Создаем файл

```
mc [dmytrymayorov@vbox]:~/work/arch-pc/lab06
variant.asm  [-] 3 L: [ 1+24 25/ 25] *(440 / 449b) 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 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
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюМС 10Выход
```

Рис. 3.19: Открываем файл и заполняем его в соответствии с листингом

```
dmytrymayorov@vbox:~/work/arch-pc/lab06$ nasm -f elf variant.asm
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132243107
Ваш вариант: 20
dmytrymayorov@vbox:~/work/arch-pc/lab06$
```

Рис. 3.20: Создаем исполняемый файл и запускаем его

4 Ответы на вопросы

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 `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`.

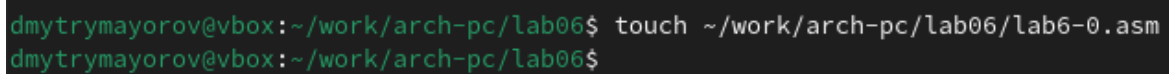
4 Строка `“xor edx,edx”` обнуляет регистр `edx` перед выполнением деления. Строка `“mov ebx,20”` загружает значение 20 в регистр `ebx`. Строка `“div ebx”` выполняет деление регистра `eax` на значение регистра `ebx` с сохранением частного в регистре `eax` и остатка в регистре `edx`.

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.

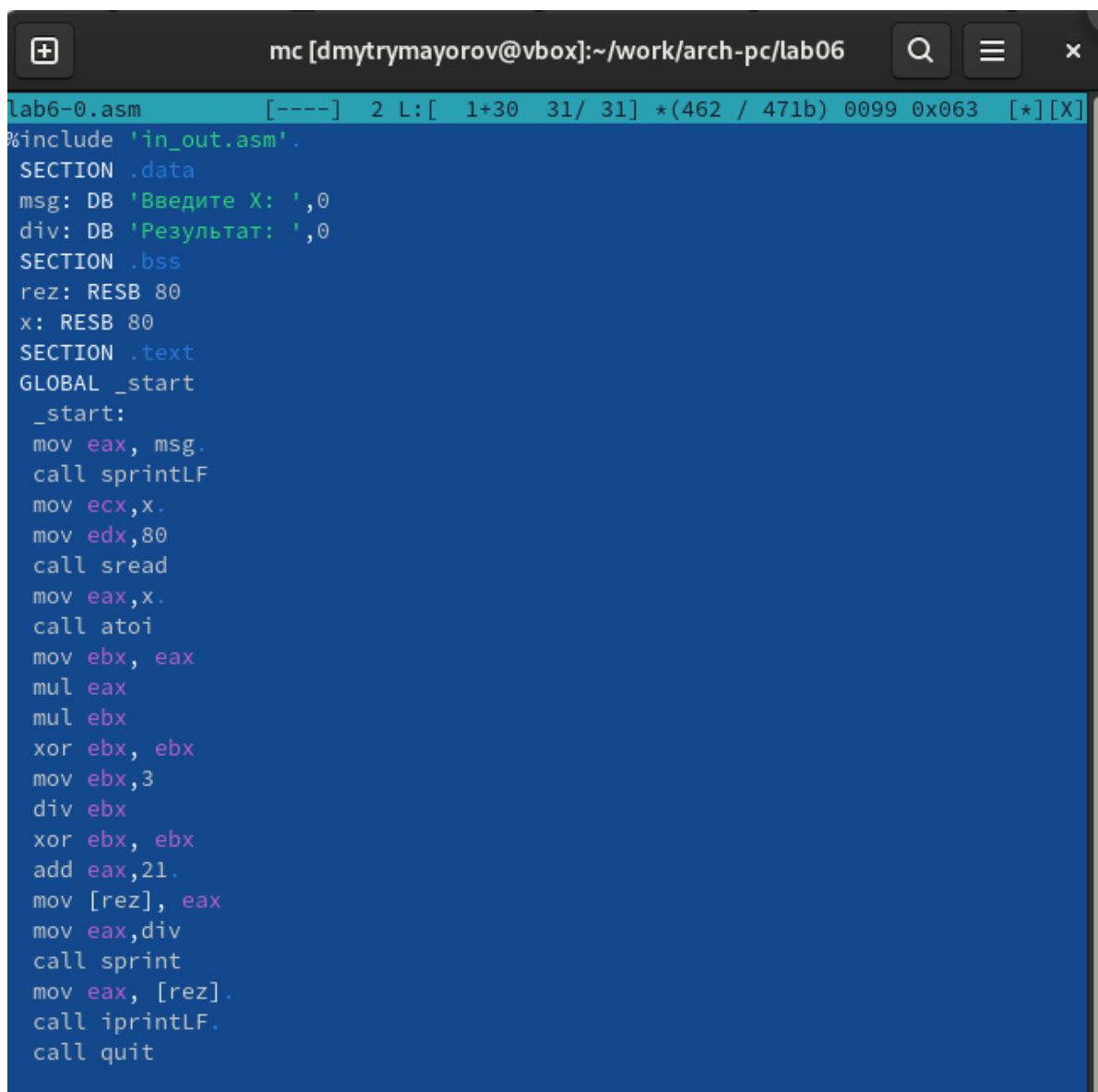
5 Задание для самостоятельной работы

A terminal window with a dark background and green text. It shows a user named dmytrymayorov@vbox at the directory ~/work/arch-pc/lab06. The user enters the command 'touch ~/work/arch-pc/lab06/lab6-0.asm' and the prompt returns.

```
dmytrymayorov@vbox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-0.asm
dmytrymayorov@vbox:~/work/arch-pc/lab06$
```

Рис. 5.1: Создаем новый файл в каталоге

Открываем его и заполняем, чтобы решалось выражение $f(x) = x^3 * 1/3 + 21$



```
mc [dmytrymayorov@vbox]:~/work/arch-pc/lab06
lab6-0.asm [----] 2 L: [ 1+30 31/ 31] *(462 / 471b) 0099 0x063 [*] [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 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. 5.2: Заполняем файл

```
dmytrymayorov@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-0.asm
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-0 lab6-0.o
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ./lab6-0
Введите X:
1
Результат: 21
dmytrymayorov@vbox:~/work/arch-pc/lab06$ mc
dmytrymayorov@vbox:~/work/arch-pc/lab06$ ./lab6-0
Введите X:
3
Результат: 30
dmytrymayorov@vbox:~/work/arch-pc/lab06$
```

Рис. 5.3: Запускаем программу и проверяем ее работу для $x=1$ и $x=3$

6 Выводы

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

Список литературы