

Отчёта по лабораторной работе 6

Освоение арифметических инструкций языка ассемблера NASM.

Жуковский Илья Сергеевич НБИбд-01-22

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	23

Список иллюстраций

3.1	Пример программы	8
3.2	Работа программы	9
3.3	Пример программы	10
3.4	Работа программы	11
3.5	Пример программы	12
3.6	Работа программы	12
3.7	Пример программы	13
3.8	Работа программы	13
3.9	Работа программы	14
3.10	Пример программы	15
3.11	Работа программы	15
3.12	Пример программы	16
3.13	Работа программы	17
3.14	Пример программы	18
3.15	Работа программы	19
3.16	Пример программы	21
3.17	Работа программы	22

Список таблиц

1 Цель работы

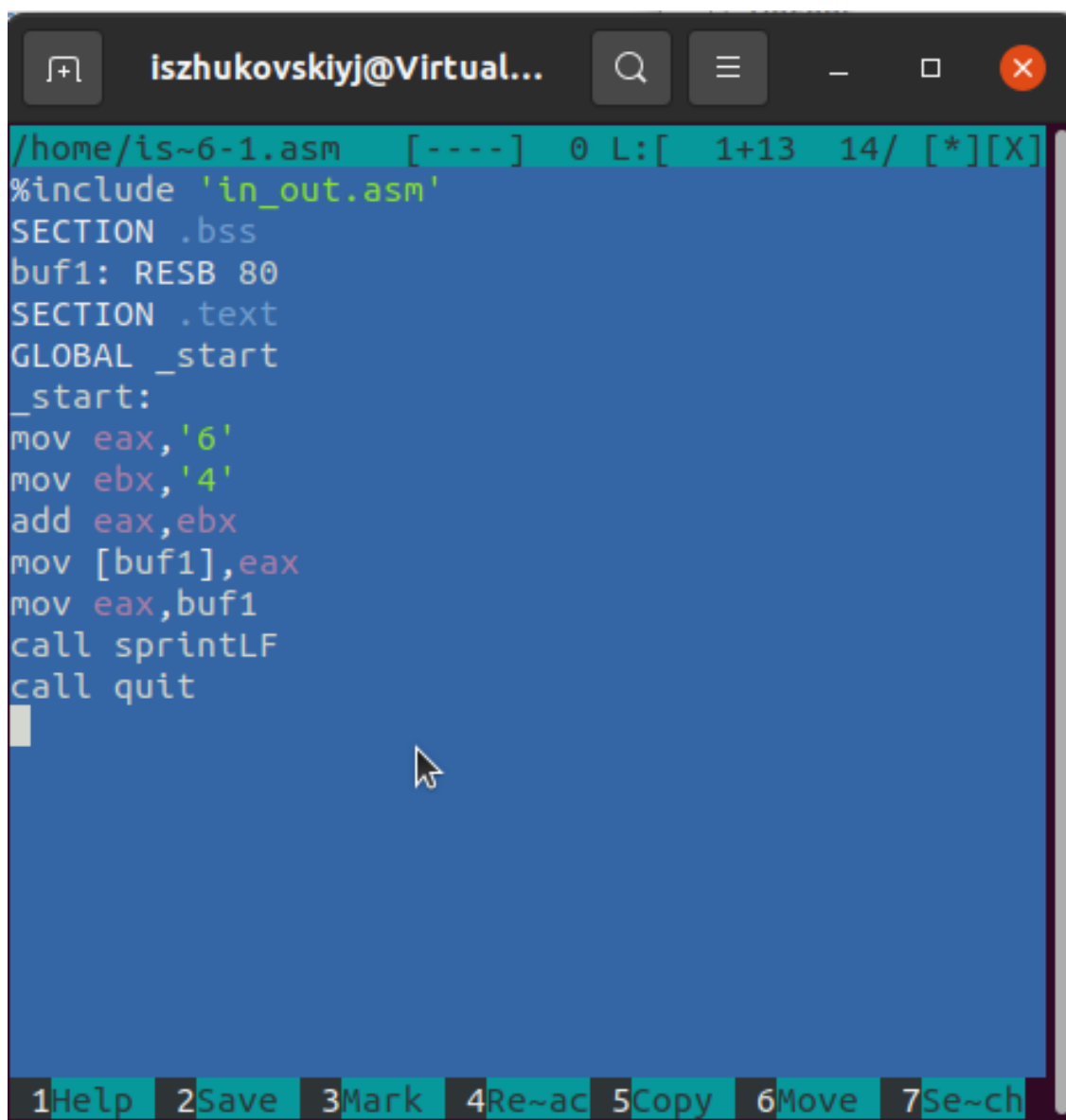
Целью работы является освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Изучите примеры программ.
2. Напишите программу вычисления выражения в соответствии с вариантом.
3. Загрузите файлы на GitHub.

3 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax. (рис. 3.1, 3.2)

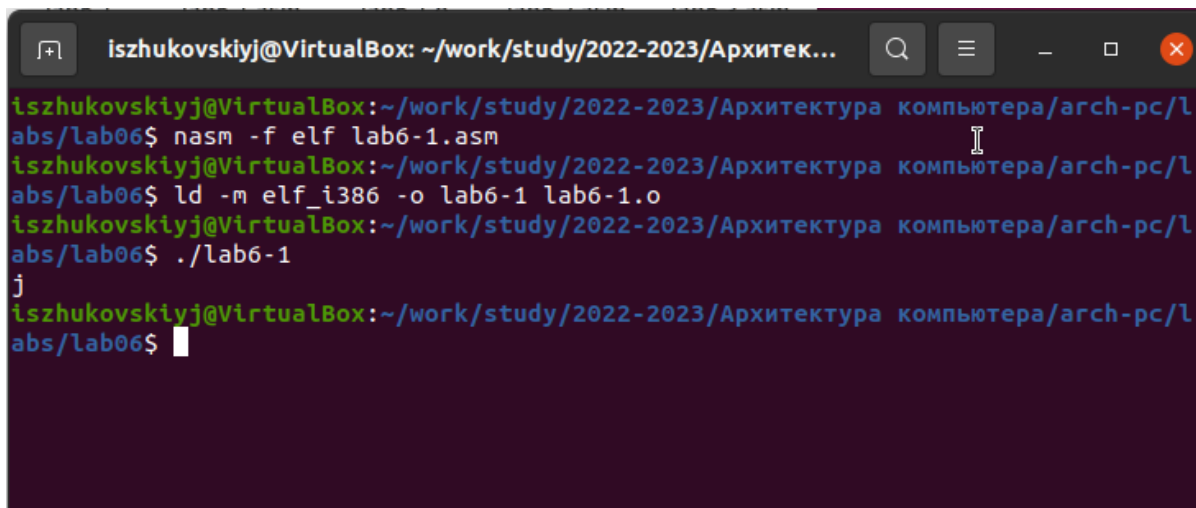


The image shows a screenshot of a code editor window. The title bar at the top reads 'iszhukovskiyj@Virtual...'. The editor area has a blue background and contains the following assembly code:

```
/home/is~6-1.asm  [ - - - ]  0  L:[  1+13  14/  [*][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
```

A mouse cursor is visible over the code. At the bottom of the window, there is a toolbar with buttons labeled: 1Help, 2Save, 3Mark, 4Re~ac, 5Copy, 6Move, 7Se~ch.

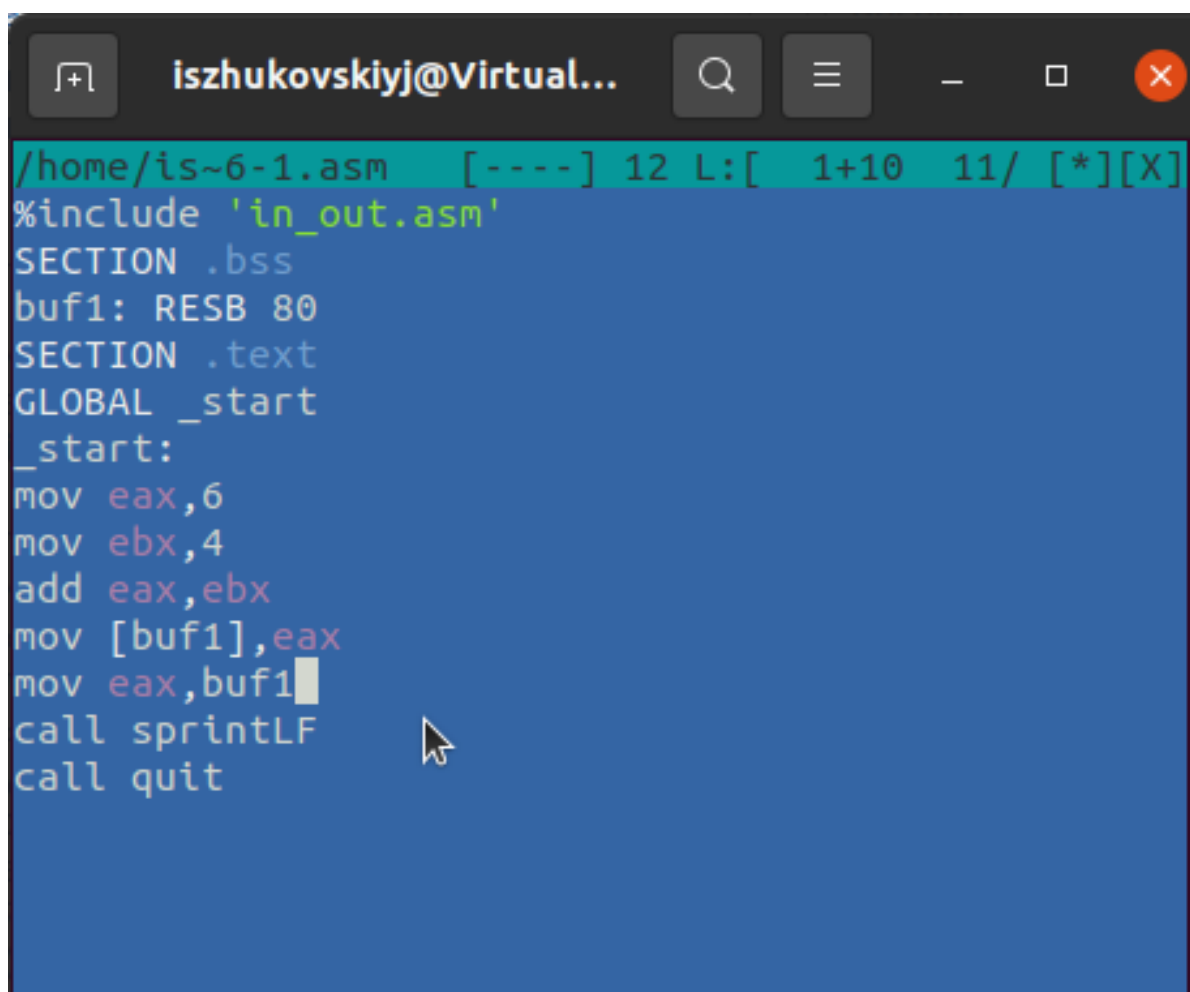
Рис. 3.1: Пример программы



```
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-1.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-1
j
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$
```

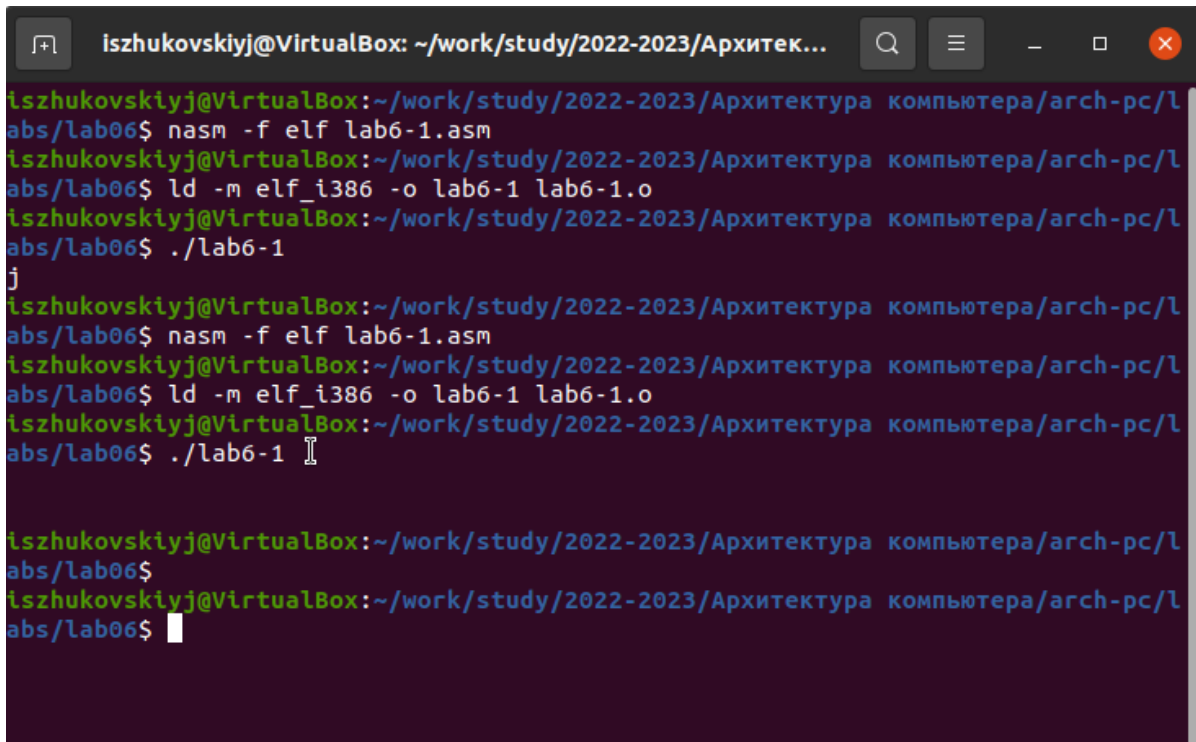
Рис. 3.2: Работа программы

3. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 1) следующим образом: (рис. 3.3, 3.4)

A screenshot of a terminal window with a dark background. The title bar at the top shows the username 'iszhukovskiyj@Virtual...' and standard window controls. The terminal displays assembly code for a program. The code includes a header file, defines a buffer, and contains a main function that calculates a sum and prints it. The code is as follows:

```
/home/is~6-1.asm [----] 12 L:[ 1+10 11/ [*]][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. 3.3: Пример программы

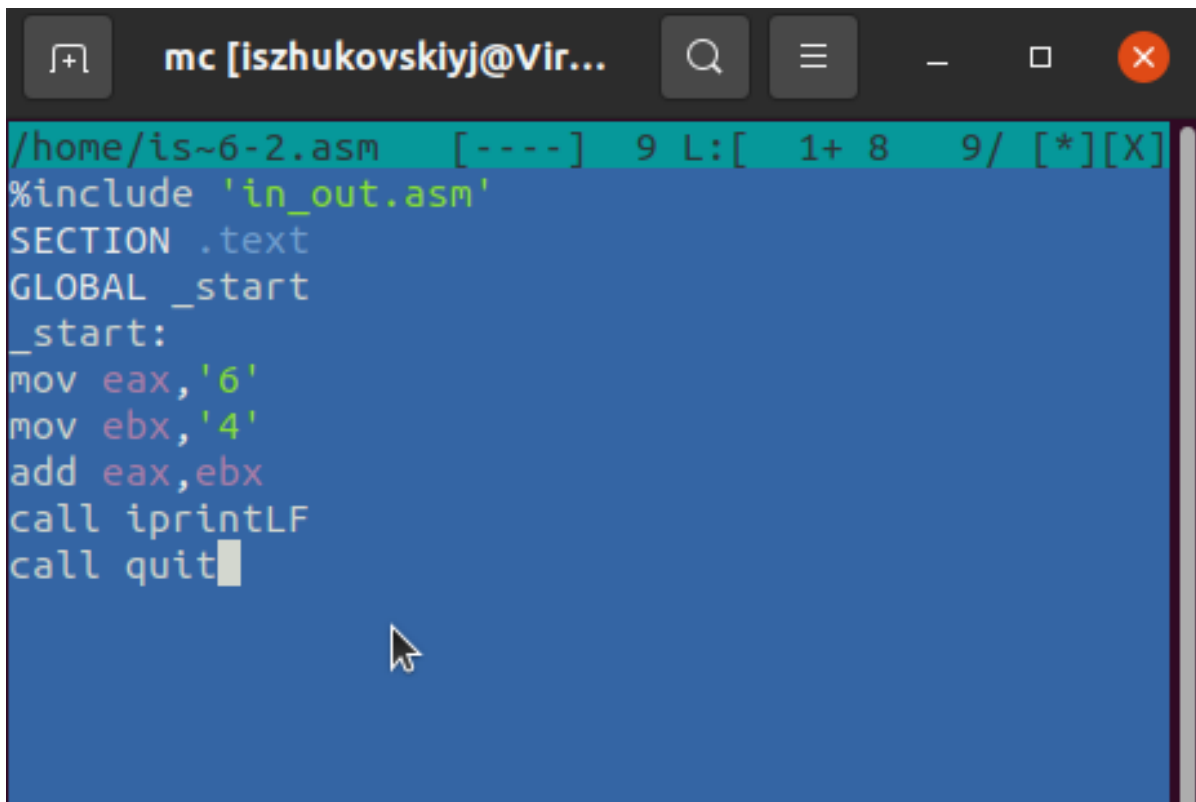
A screenshot of a terminal window titled 'iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитек...'. The terminal shows a series of commands and their outputs. The user is in the directory ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06. The commands executed are: 'nasm -f elf lab6-1.asm', 'ld -m elf_i386 -o lab6-1 lab6-1.o', and './lab6-1'. The output of the first two commands is not shown. The output of the third command is a single character 'j'. The terminal window has a dark background with light-colored text. The window title bar includes standard Linux window controls (minimize, maximize, close) and a search icon.

```
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитек...
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ./lab6-1
j
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-1.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ./lab6-1
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.4: Работа программы

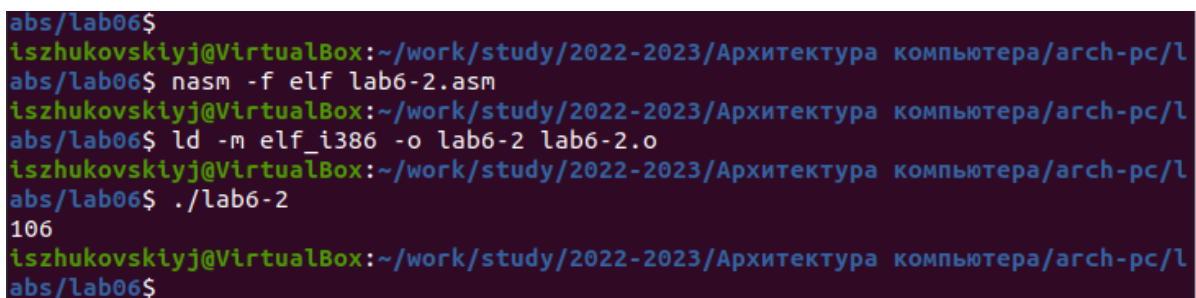
Никакой символ не виден, но он есть. Это возврат каретки LF.

4. Как отмечалось выше, для работы с числами в файле in_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 7.1 с использованием этих функций. (рис. 3.5, 3.6)



```
mc [iszhukovskiyj@Vir...
/home/is~6-2.asm [----] 9 L:[ 1+ 8 9/ [*][X]
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 3.5: Пример программы



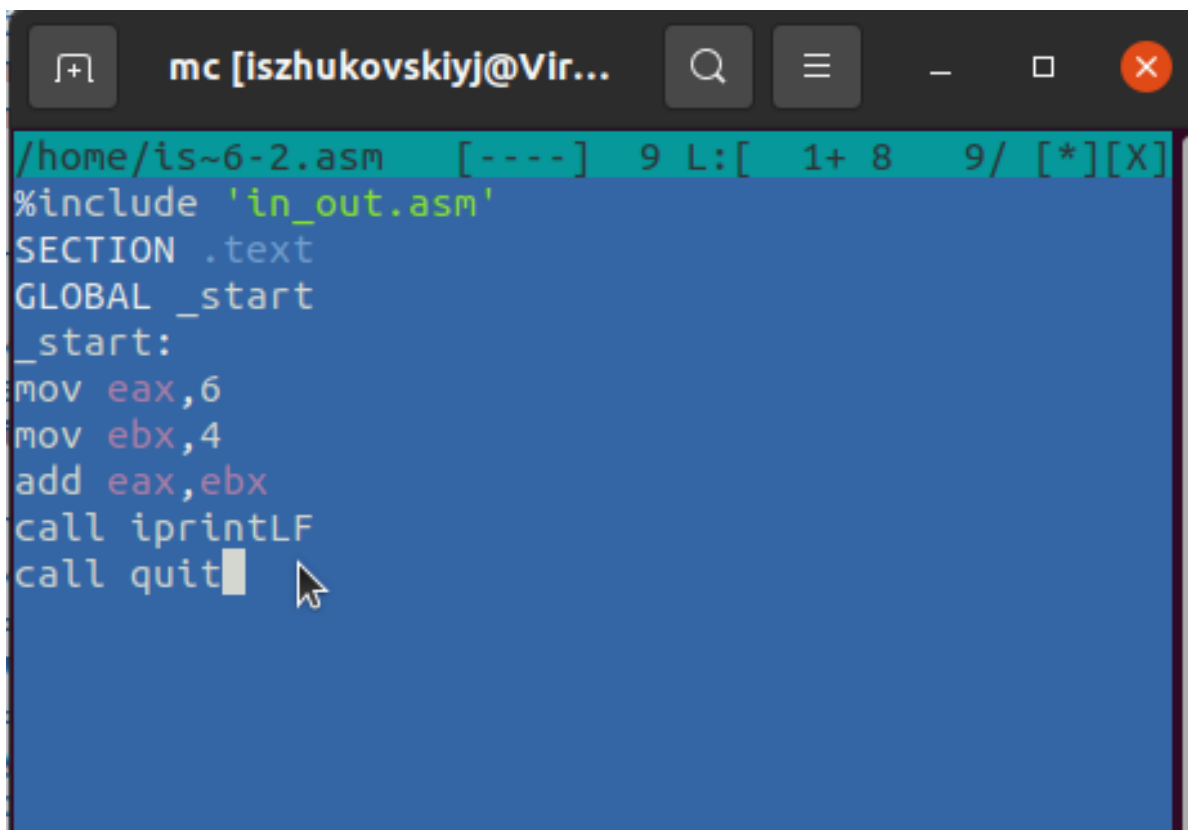
```
abs/lab06$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ nasm -f elf lab6-2.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ./lab6-2
106
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$
```

Рис. 3.6: Работа программы

В результате работы программы мы получим число 106. В данном случае, как и в первом, команда `add` складывает коды символов '6' и '4' ($54+52=106$). Однако, в отличие от программы из листинга 7.1, функция `iprintLF` позволяет вывести число, а не символ, кодом которого является это число.

5. Аналогично предыдущему примеру изменим символы на числа. (рис. 3.7, 3.8)

Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы? – получили число 10



```
mc [iszhukovskiyj@Vir...
/home/is~6-2.asm [----] 9 L:[ 1+ 8 9/ [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

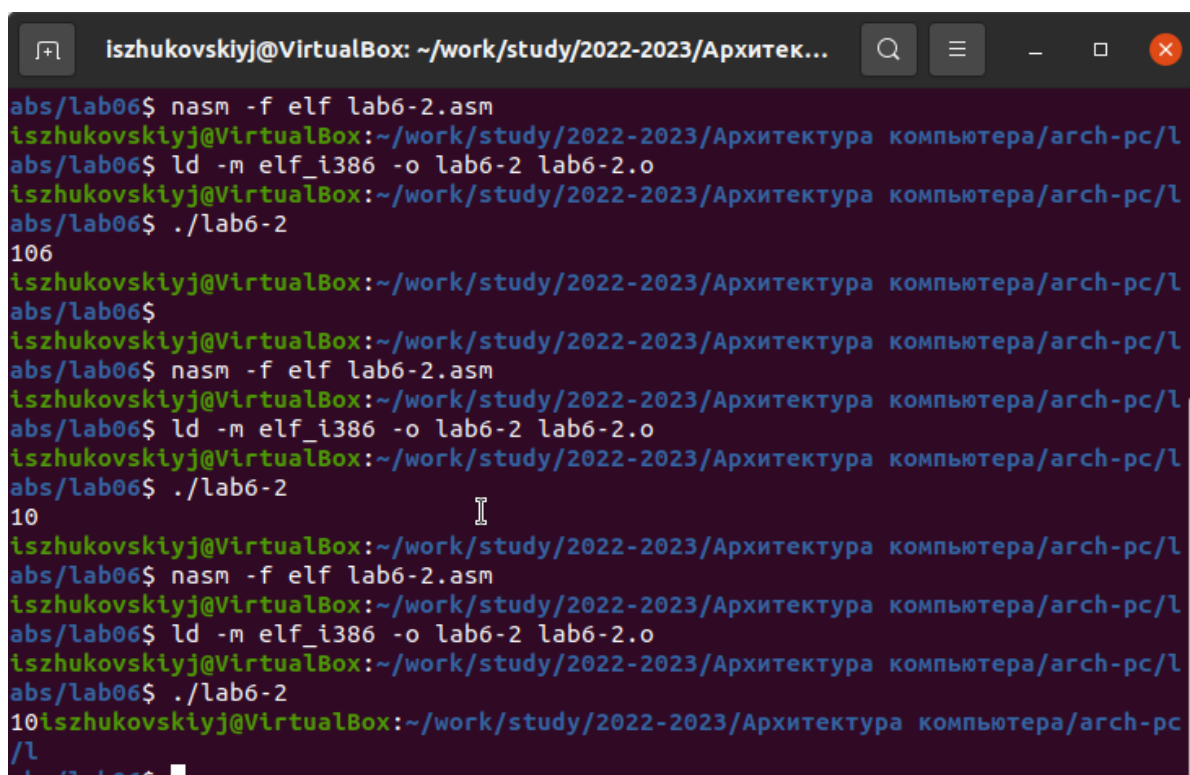
Рис. 3.7: Пример программы



```
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
106
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-2.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ./lab6-2
10
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$
```

Рис. 3.8: Работа программы

Замените функцию `iprintln` на `iprint`. Создайте исполняемый файл и запустите его. Чем отличается вывод функций `iprintln` и `iprint`? - Вывод отличается что нет переноса строки. (рис. 3.9)



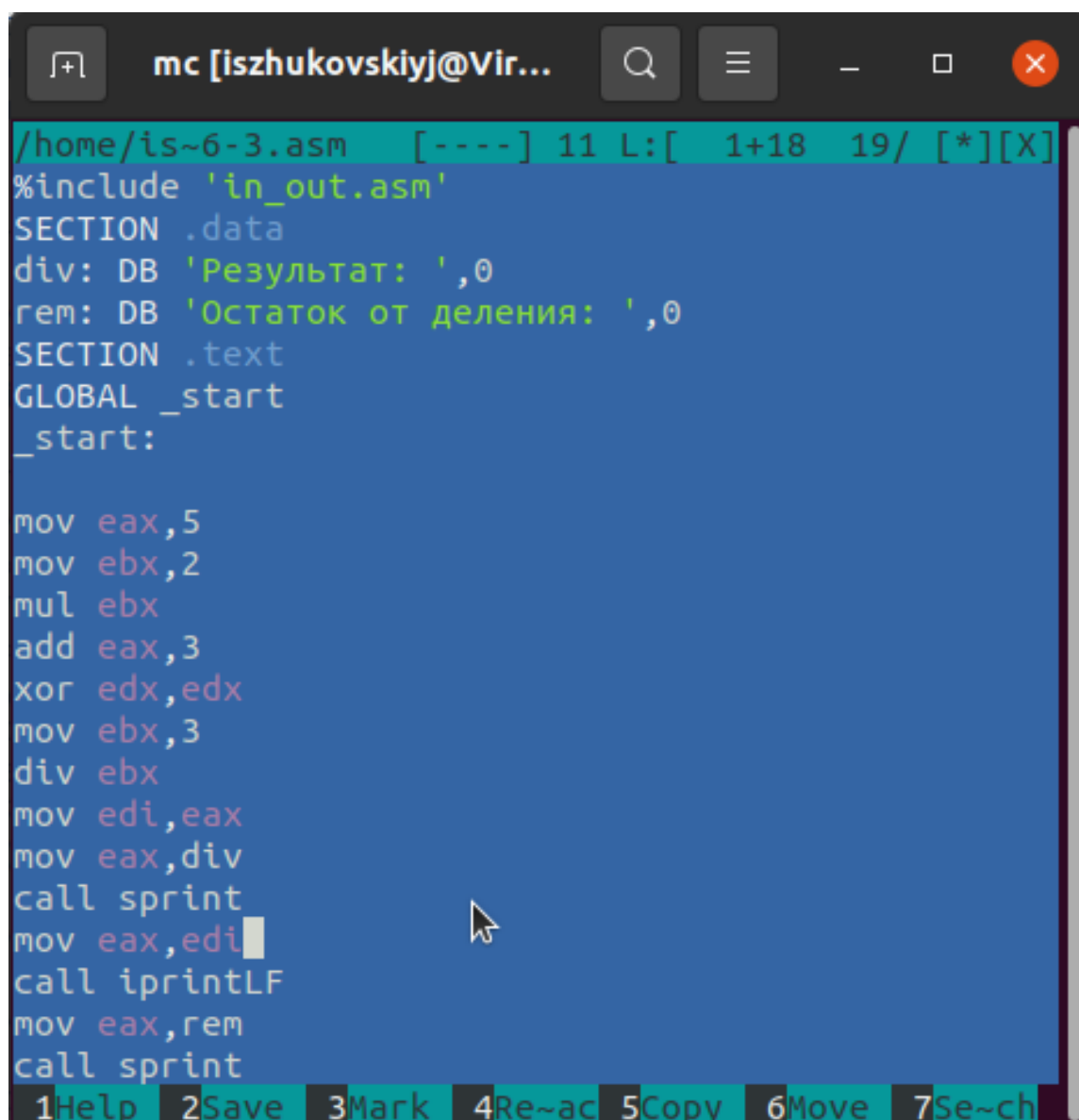
```
abs/lab06$ nasm -f elf lab6-2.asm
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ./lab6-2
106
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ nasm -f elf lab6-2.asm
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ./lab6-2
10
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ nasm -f elf lab6-2.asm
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ./lab6-2
10iszhukovskiyj@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc
/l
```

Рис. 3.9: Работа программы

6. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения

$$f(x) = (5 * 2 + 3) / 3$$

. (рис. 3.10, рис. 3.11)

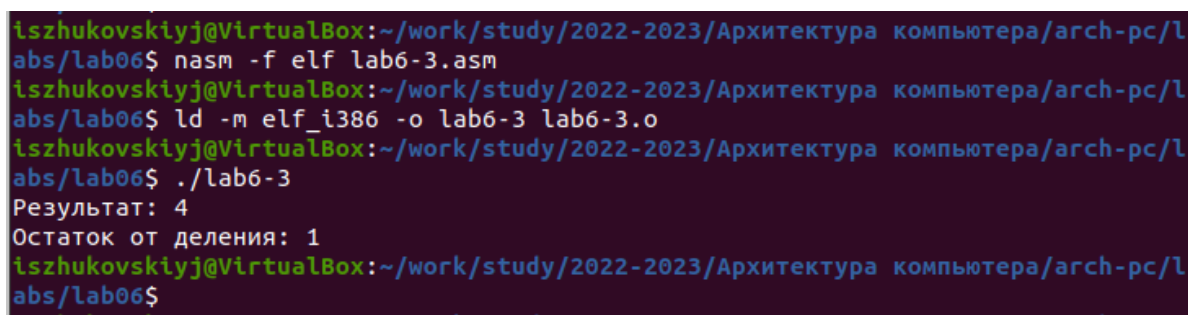


The screenshot shows a text editor window titled 'mc [iszhukovskiyj@Vir...'. The code is written in assembly language and includes comments in Russian. It defines two data sections: 'div' with the string 'Результат: ',0 and 'rem' with the string 'Остаток от деления: ',0. The text section starts at '_start:' and contains instructions to calculate 5 divided by 2. The result is stored in 'eax' (4) and 'edx' (1). The program then prints the result and remainder using 'sprint' and 'iprintLF' functions. A menu bar at the bottom includes options like '1Help', '2Save', '3Mark', '4Re~ac', '5Copy', '6Move', and '7Se~ch'.

```
/home/is~6-3.asm [----] 11 L:[ 1+18 19/ [*][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
```

Рис. 3.10: Пример программы



The screenshot shows a terminal window with the following commands and output: 'nasm -f elf lab6-3.asm' to assemble the code, 'ld -m elf_i386 -o lab6-3 lab6-3.o' to link it, and './lab6-3' to run it. The output shows the program's results: 'Результат: 4' and 'Остаток от деления: 1'.

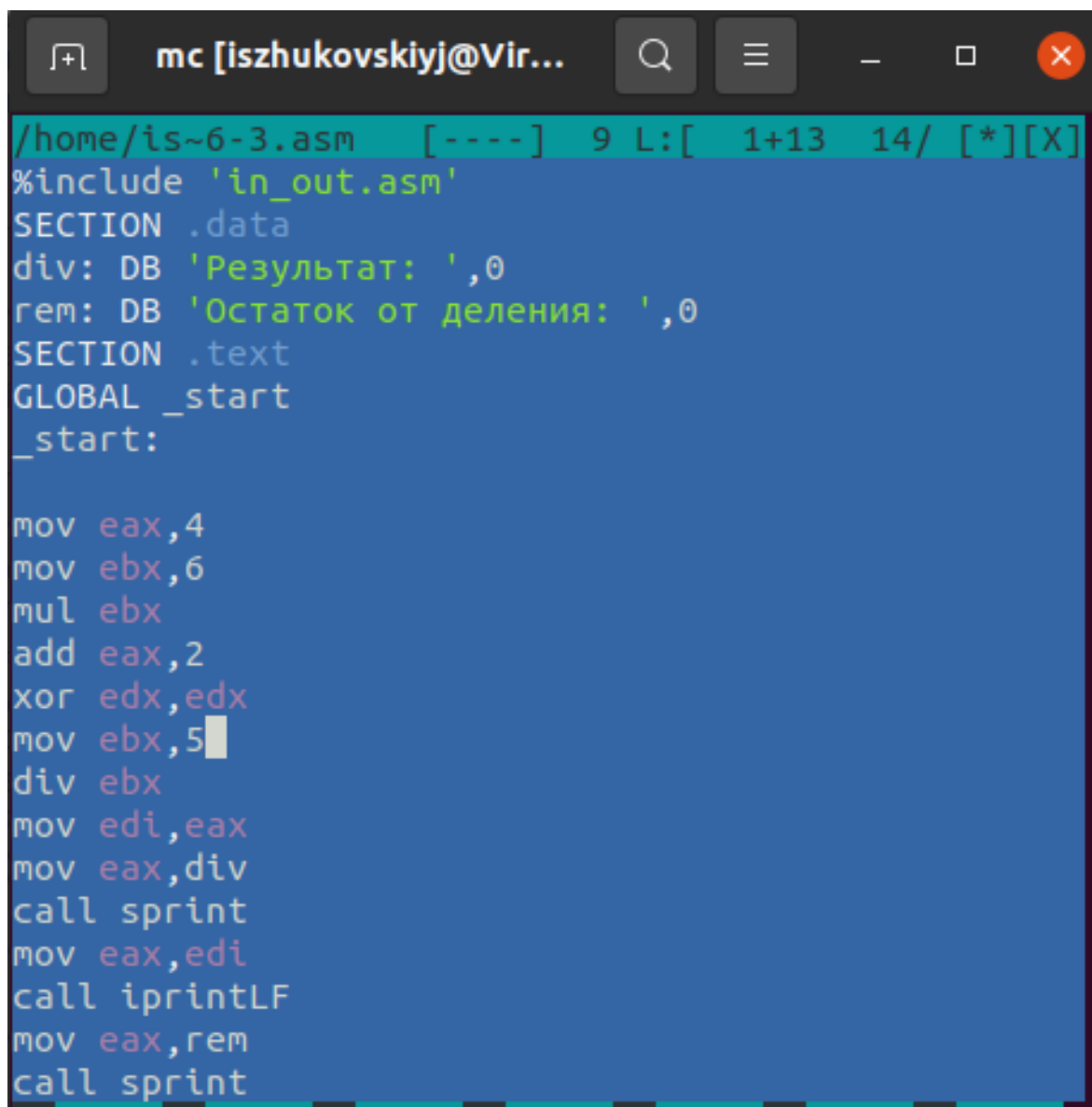
```
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab6-3$ nasm -f elf lab6-3.asm
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab6-3$ ld -m elf_i386 -o lab6-3 lab6-3.o
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab6-3$ ./lab6-3
Результат: 4
Остаток от деления: 1
iszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab6-3$
```

Рис. 3.11: Работа программы

Измените текст программы для вычисления выражения

$$f(x) = (4 * 6 + 2) / 5$$

. Создайте исполняемый файл и проверьте его работу. (рис. 3.12, рис. 3.13)

A screenshot of a code editor window titled 'mc [iszhukovskiyj@Vir...'. The editor shows assembly code for a program that calculates the expression (4*6+2)/5. The code includes comments in Russian for the output labels. The assembly instructions are: 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. The code is color-coded: keywords in blue, registers and constants in red, and strings in green.

```
/home/is~6-3.asm [----] 9 L:[ 1+13 14/ [*][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
```

Рис. 3.12: Пример программы

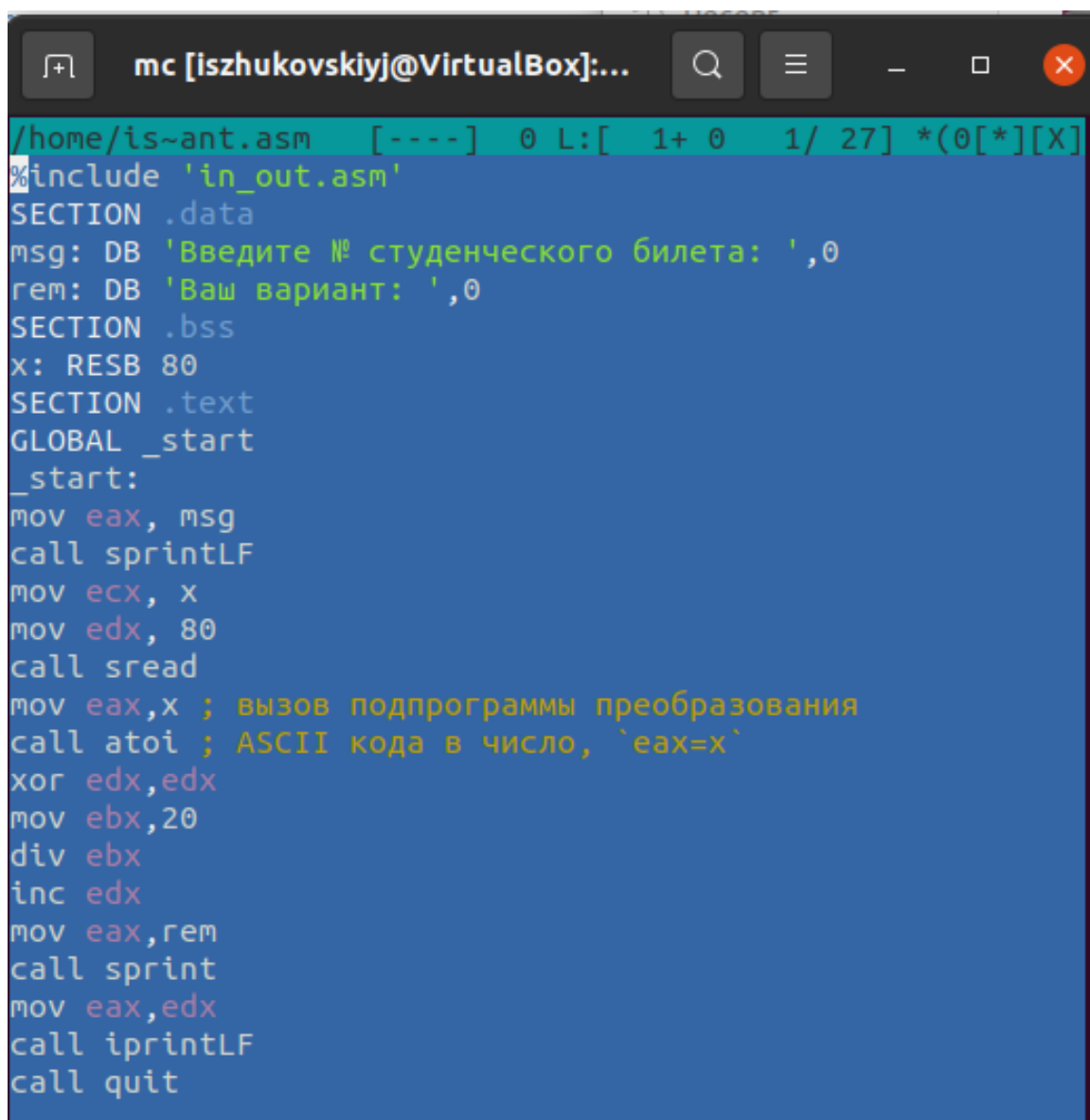

```

lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ nasm -f elf lab6-3.asm
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ nasm -f elf lab6-3.asm
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l
abs/lab06$
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/l

```

Рис. 3.13: Работа программы

7. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму: (рис. 3.14, рис. 3.15)



```
mc [iszhukovskiyj@VirtualBox]:...  
/home/is~ant.asm [----] 0 L:[ 1+ 0 1/ 27] *(0[*])[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 ; вызов подпрограммы преобразования  
call atoi ; ASCII кода в число, `eax=x`  
xor edx, edx  
mov ebx, 20  
div ebx  
inc edx  
mov eax, rem  
call sprintf  
mov eax, edx  
call iprintLF  
call quit
```

Рис. 3.14: Пример программы

```

lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ nasm -f elf variant.asm
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132222836
Ваш вариант: 17
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06$

```

Рис. 3.15: Работа программы

- Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? – `mov eax,rem` – перекладывает в регистр значение переменной с фразой ‘Ваш вариант:’ `call sprint` – вызов подпрограммы вывода строки
- Для чего используются следующие инструкции? `nasm mov ecx, x mov edx, 80 call sread`

Считывает значение студбилета в переменную X из консоли

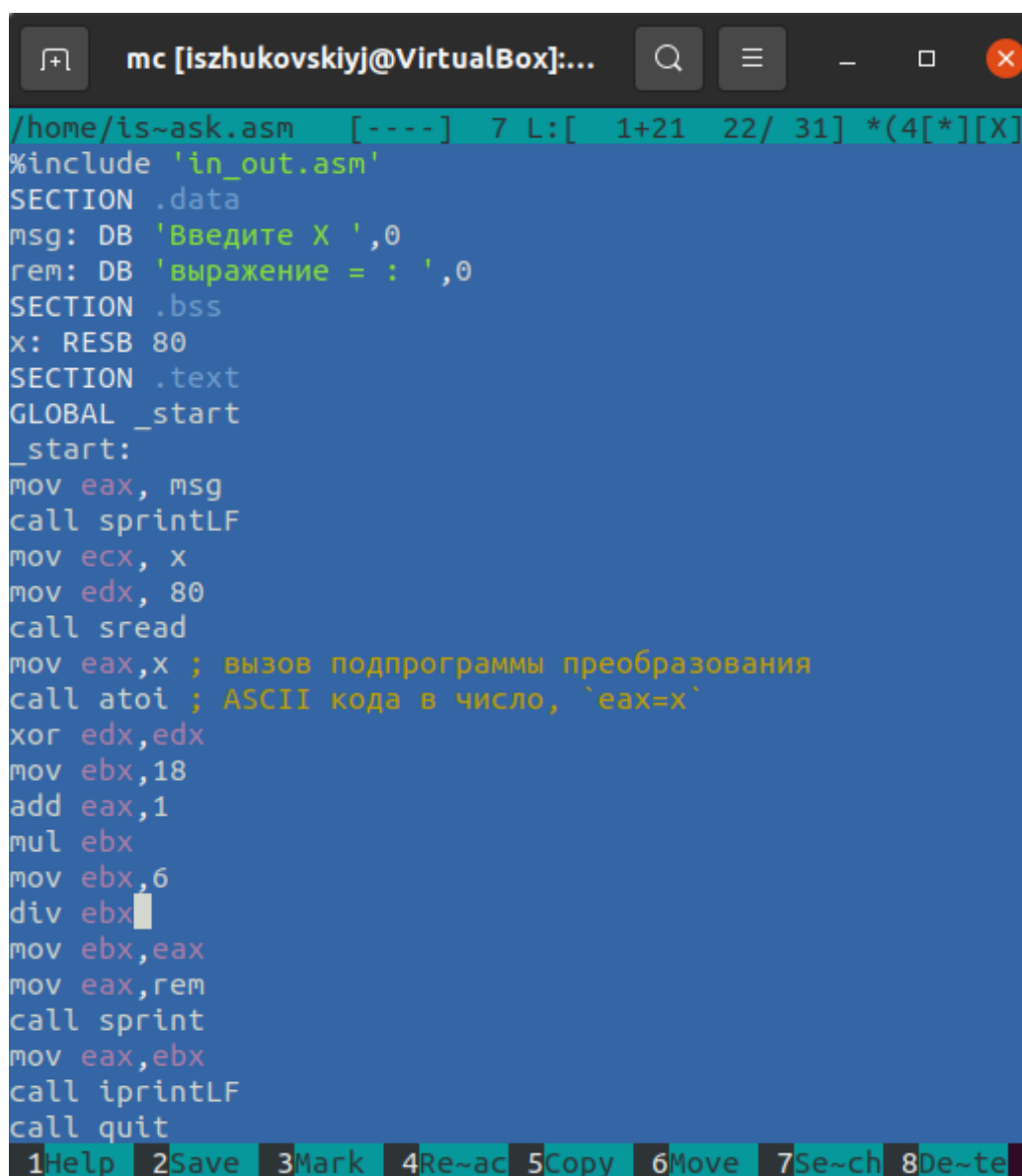
- Для чего используется инструкция “`call atoi`”? - эта подпрограмма переводит введенные символы в числовой формат
- Какие строки листинга 7.4 отвечают за вычисления варианта? `xor edx,edx mov ebx,20 div ebx`
- В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”? 1 байт AH 2 байта DX 4 байта EDX – наш случай
- Для чего используется инструкция “`inc edx`”? по формуле вычисления варианта нужно прибавить единицу
- Какие строки листинга 7.4 отвечают за вывод на экран результата вычисления? `mov eax,edx` – результат перекладывается в регистр `eax` `call iprintLF` – вызов подпрограммы вывода

8. Написать программу вычисления выражения $y = f(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат вычислений. Вид функции $f(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 из 6.3. (рис. 3.16, рис. 3.17)

Получили вариант 17 -

$$18(x + 1)/6$$

для $x=3$ и 1



```
mc [iszhukovskiyj@VirtualBox]:...
/home/is~ask.asm [----] 7 L:[ 1+21 22/ 31] *(4[*])[X]
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите X ',0
rem: DB 'выражение = : ',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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx, edx
mov ebx, 18
add eax, 1
mul ebx
mov ebx, 6
div ebx
mov ebx, eax
mov eax, rem
call sprintf
mov eax, ebx
call iprintfLF
call quit
1Help 2Save 3Mark 4Re~ac 5Copy 6Move 7Se~ch 8De~te
```

Рис. 3.16: Пример программы

```
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf task.asm
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o task task.o
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$ ./task
Введите X
3
выражение = : 12
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$ ./task
Введите X
1
выражение = : 6
lszhukovskiyj@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис. 3.17: Работа программы

4 Выводы

Изучили работу с арифметическими операциями