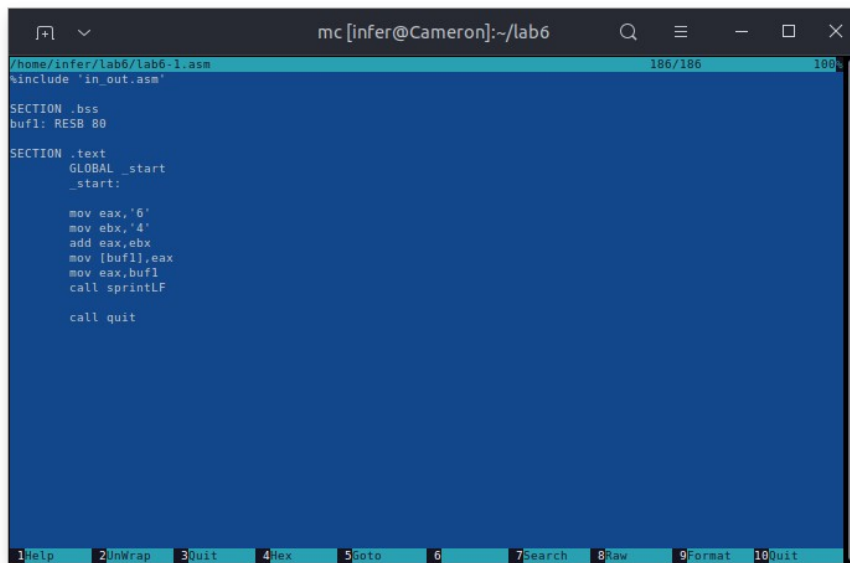


Лабораторная работа №6. Арифметические операции в NASM

Цель работы: Освоение арифметических инструкций языка ассемблера NASM.

Ход работы.

Создадим программу вывода значения регистра `eax` с исходным кодом, показанным на рисунке 1.



```
mc [infer@Cameron]:~/lab6
/home/infer/lab6/lab6-1.asm 186/186 100%
#include 'in_out.asm'

SECTION .bss
buf1: RESB 80

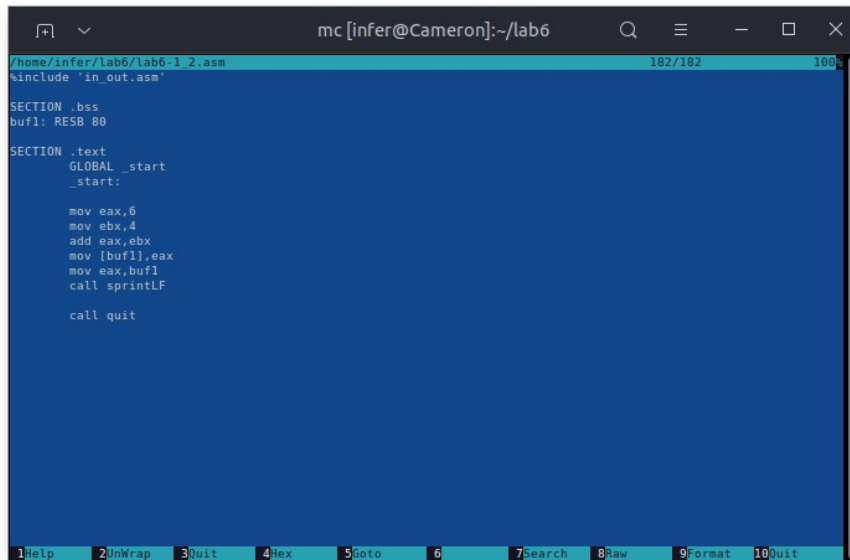
SECTION .text
GLOBAL _start
_start:

    mov eax, '6'
    mov ebx, '4'
    add eax, ebx
    mov [buf1], eax
    mov eax, buf1
    call sprintf
    call quit

1 Help 2 UnWrap 3 Quit 4 Hex 5 Goto 6 7 Search 8 Raw 9 Format 10 Quit
```

Рисунок 1 — Исходный код программы lab6-1

В ней в регистры `eax` и `ebx` записываются символьные константы, которые затем складываются и результат выводится на экран. Также создадим аналогичную программу, в которой идет работа с числовыми константами (рисунок 2)



The screenshot shows a text editor window titled 'mc [infer@Cameron]:~/lab6'. The code is for 'lab6-1_2.asm' and includes the following assembly instructions:

```
~/home/infer/lab6/lab6-1_2.asm 182/182 100%
#include 'in_out.asm'

SECTION .bss
buf1: RESB 80

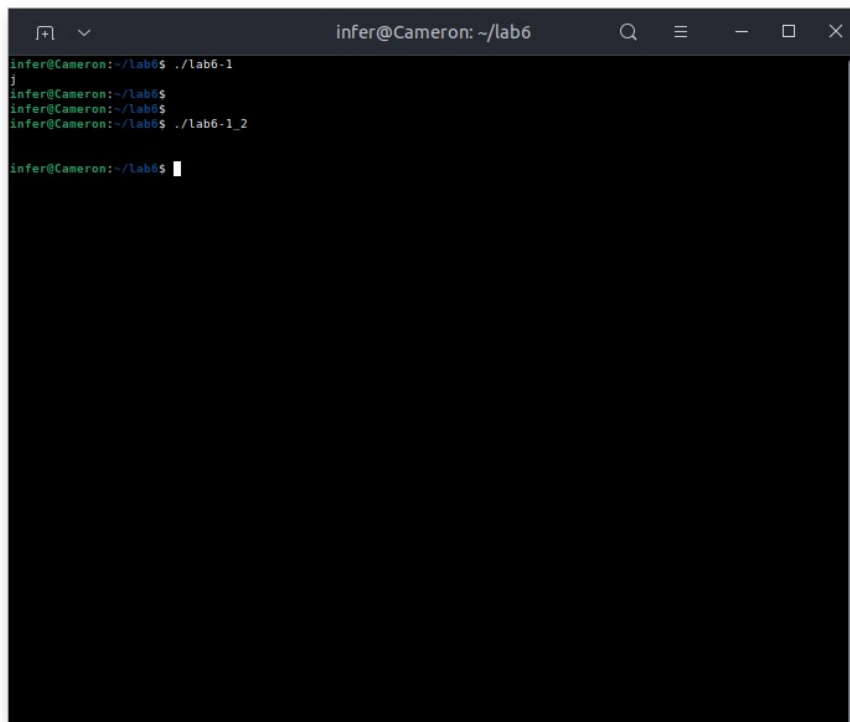
SECTION .text
GLOBAL _start
_start:

    mov eax,6
    mov ebx,4
    add eax,ebx
    mov [buf1],eax
    mov eax,buf1
    call sprintf
    call quit
```

The editor has a status bar at the bottom with buttons: 1Help, 2InWrap, 3Quit, 4Hex, 5Goto, 6, 7Search, 8Raw, 9Format, 10Quit.

Рисунок 2 — Исходный код программы lab6-1_2

Результат выполнения этих программ показан на рисунке 3.



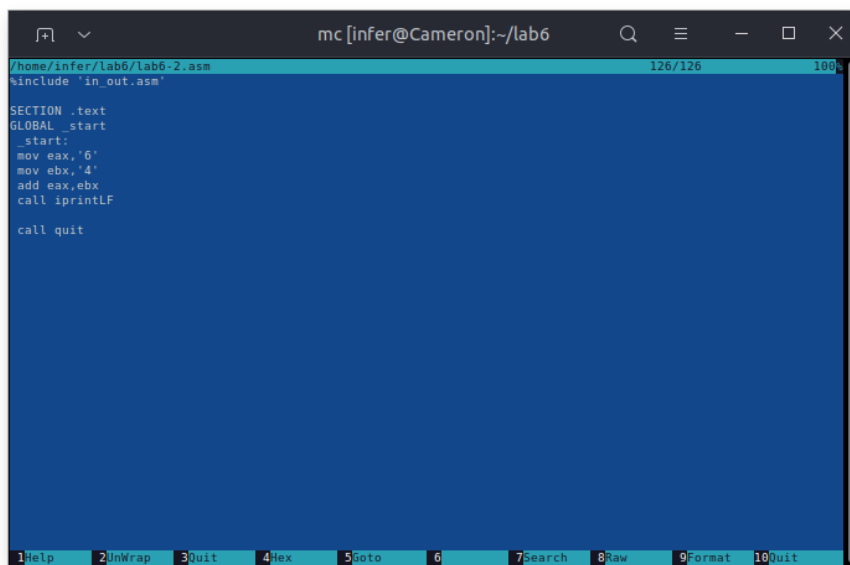
The screenshot shows a terminal window titled 'infer@Cameron: ~/lab6'. The execution sequence is as follows:

```
infer@Cameron:~/lab6$ ./lab6-1
j
infer@Cameron:~/lab6$
infer@Cameron:~/lab6$
infer@Cameron:~/lab6$ ./lab6-1_2
infer@Cameron:~/lab6$
```

Рисунок 3 — Результат выполнения программ

В первом случае выводится символ j, во втором символ перевода строки. Это связано с тем, что NASM при сложении символьных констант интерпретирует их как соответствующие числа из таблицы ASCII. Аналогично происходит при выводе на экран (сумма 6 и 4 равна 10, что в таблице ASCII соответствует символу перевода строки).

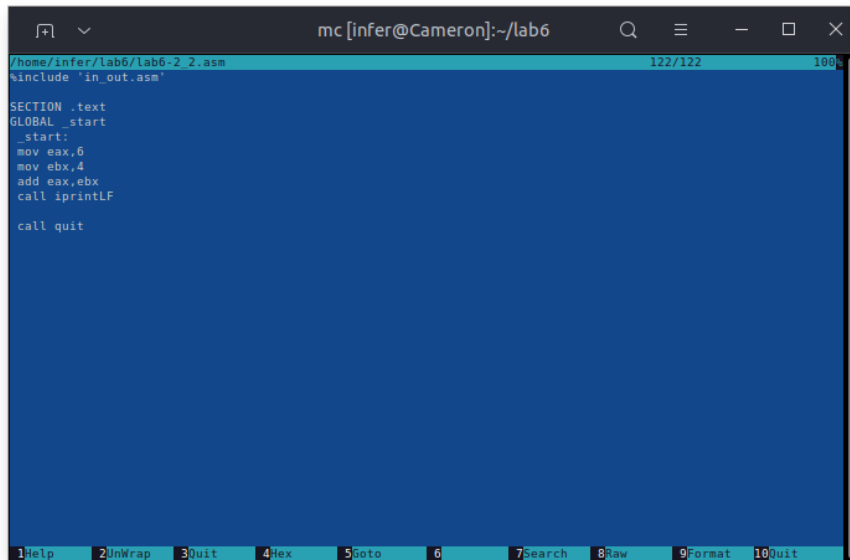
При этом, использование функций `iprintLF` и `iprint` из файла `in_out.asm` позволяет выводить непосредственно числовые значения. Отличия этих функций заключаются в том, что первая добавляет к выводу символ переноса строки. На рисунках 4 и 5 показаны исходные коды программ, которые демонстрируют работу этих функций. На рисунке 6 показан результат работы программ.



```
mc [infer@Cameron]:~/lab6 126/126 100%
/home/infer/lab6/lab6-2.asm
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:
    mov eax,'6'
    mov ebx,'4'
    add eax,ebx
    call iprintLF
    call quit
```

Рисунок 4 — Исходный код программы lab6-2

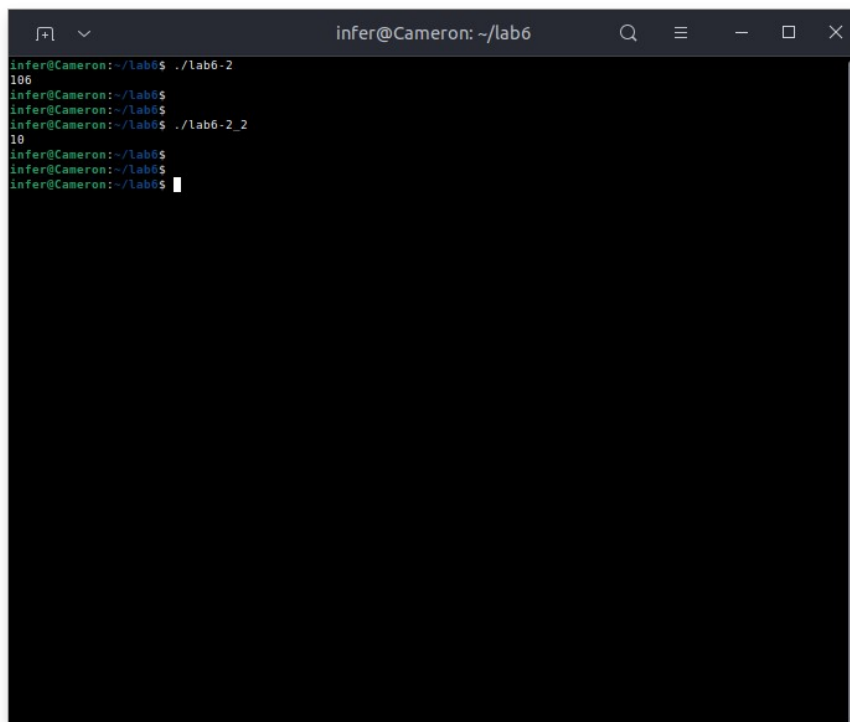


The screenshot shows a text editor window titled "mc [infer@Cameron]:~/lab6". The editor displays the assembly code for "lab6-2_2.asm". The code includes a header section, a text section, and a global start symbol. The start symbol points to a "start" label, which contains instructions to move the value 6 into the EAX register, move the value 4 into the EBX register, add EBX to EAX, call the "iprintLF" function, and finally call the "quit" function. The editor has a status bar at the bottom with various menu items like Help, InWrap, Quit, Hex, Goto, Search, Raw, Format, and Quit.

```
~/home/infer/lab6/lab6-2_2.asm 122/122 100%
#include "in_out.asm"

SECTION .text
GLOBAL _start
_start:
    mov eax,6
    mov ebx,4
    add eax,ebx
    call iprintLF
    call quit
```

Рисунок 5 — Исходный код программы lab6-2_2

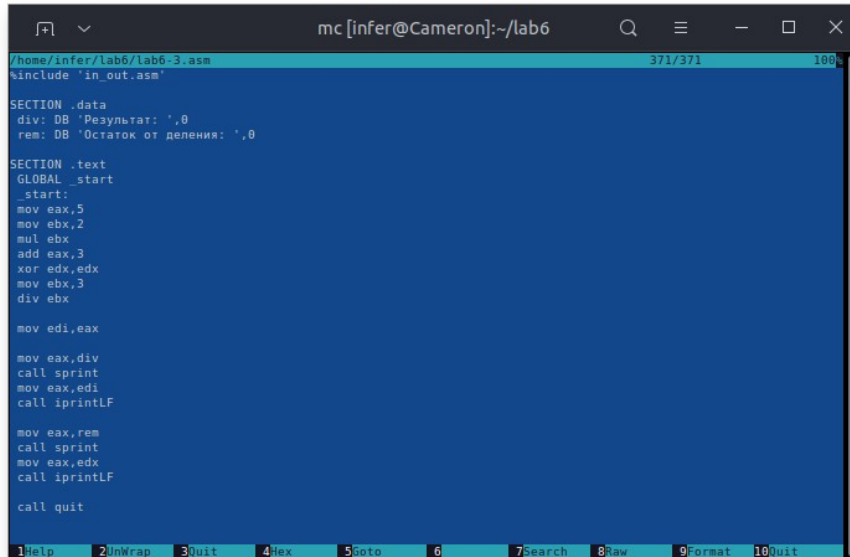


The screenshot shows a terminal window titled "infer@Cameron: ~/lab6". The terminal displays the execution of the "lab6-2_2" program. The user enters the command "infer@Cameron:~/lab6\$./lab6-2_2", and the program outputs the number "106". The user then enters the command "infer@Cameron:~/lab6\$./lab6-2_2", and the program outputs the number "10". The terminal shows the prompt "infer@Cameron:~/lab6\$" and a cursor at the end of the line.

```
infer@Cameron:~/lab6$ ./lab6-2_2
106
infer@Cameron:~/lab6$ ./lab6-2_2
10
infer@Cameron:~/lab6$
```

Рисунок 6 – Результат выполнения программ

Создадим программы для вычисления выражений $(5*2+3)/3$ и $(4*6+2)/5$. Исходные коды программ показаны на рисунках 7 и 8. Как видно из рисунков, код программ по сути идентичен, различия только в записываемых в регистры исходных данных.



```
mc [infer@Cameron]:~/lab6
/home/infer/lab6/lab6-3.asm 371/371 100%
#include 'in_out.asm'

SECTION .data
div: DB 'Делитель: ',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

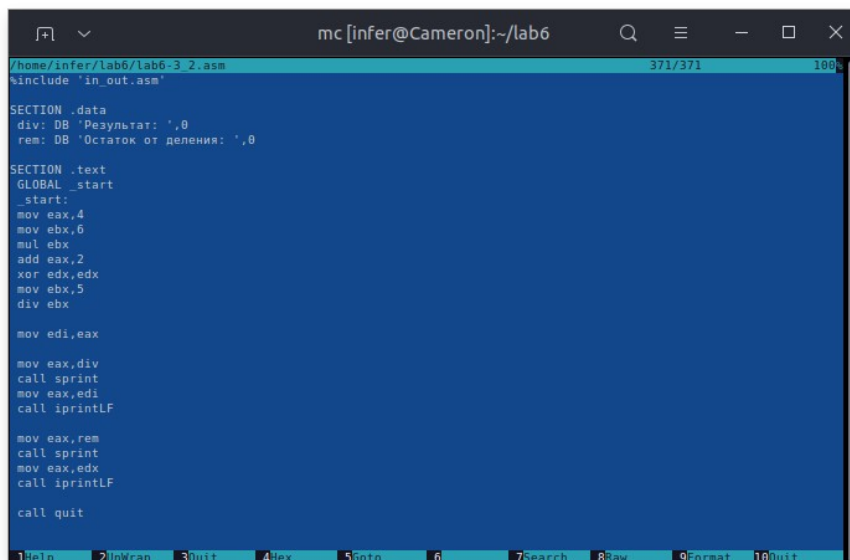
mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintfLF

mov eax,rem
call sprint
mov eax,edx
call iprintfLF

call quit
```

Рисунок 7 — Исходный код программы lab6-3



```
mc [infer@Cameron]:~/lab6
/home/infer/lab6/lab6-3_2.asm 371/371 100%
#include 'in_out.asm'

SECTION .data
div: DB 'Делитель: ',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

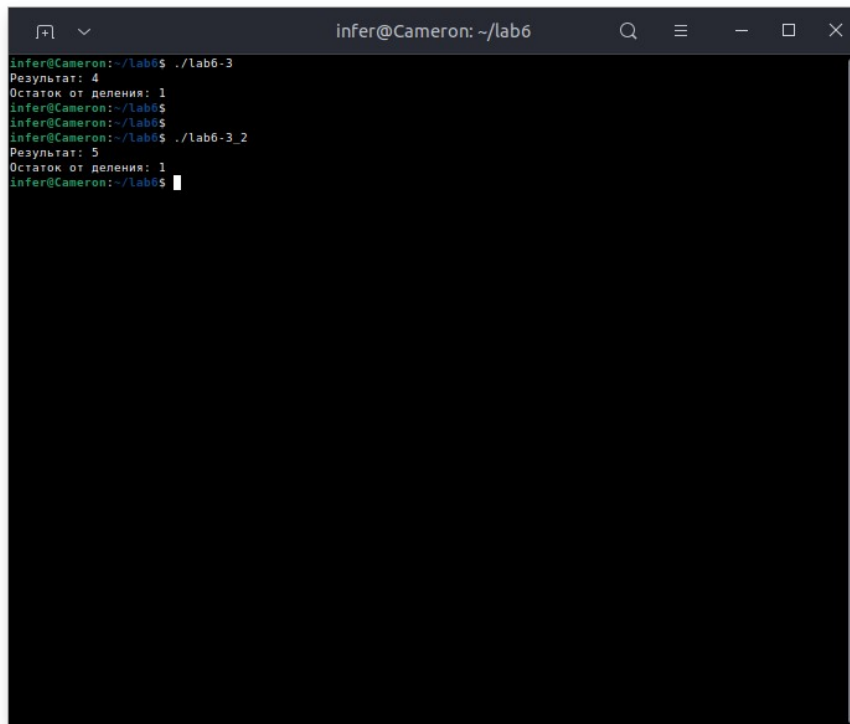
mov eax,div
call sprint
mov eax,edi
call iprintfLF

mov eax,rem
call sprint
mov eax,edx
call iprintfLF

call quit
```

Рисунок 8 — Исходный код программы lab6-3_2

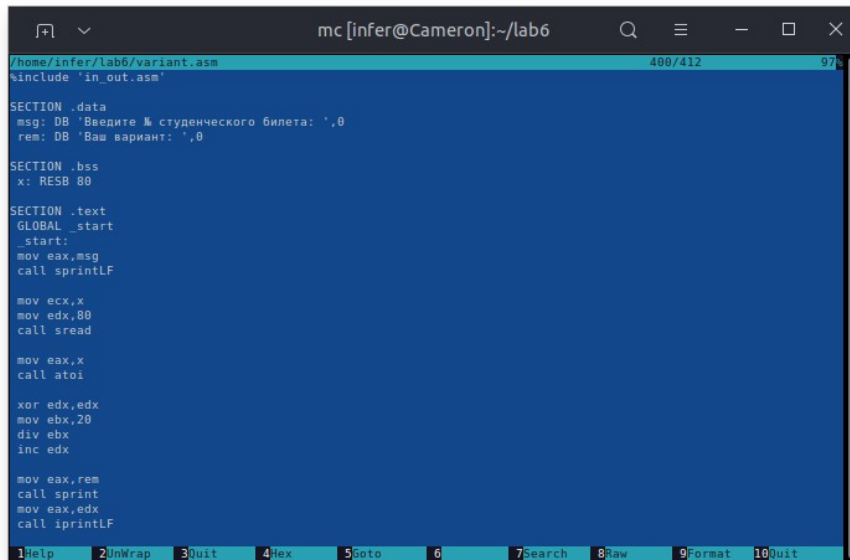
Результат выполнения программ показан на рисунке 9.

A terminal window with a dark background and light-colored text. The window title is 'infer@Cameron: ~/lab6'. The terminal shows the following commands and outputs:

```
infer@Cameron:~/lab6$ ./lab6-3
Результат: 4
Остаток от деления: 1
infer@Cameron:~/lab6$
infer@Cameron:~/lab6$
infer@Cameron:~/lab6$ ./lab6-3_2
Результат: 5
Остаток от деления: 1
infer@Cameron:~/lab6$
```

Рисунок 9 — Результат выполнения программ

Далее создадим программу для определения номера варианта по номеру зачетной книжки. Исходный код программы показан на рисунке 10, результат выполнения на рисунке 11.



```
mc [infer@Cameron]:~/lab6 400/412 97%
/home/infer/lab6/variant.asm
#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

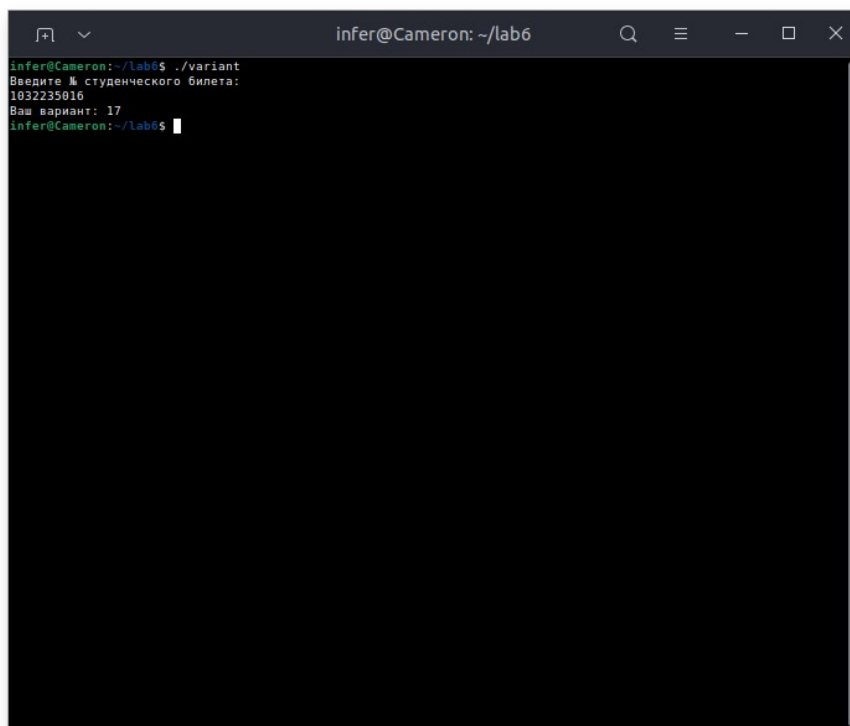
mov ecx,x
mov edx,80
call sread

mov eax,x
call atoi

xor edx,edx
mov ebx,20
div ebx
inc edx

mov eax,rem
call sprintf
mov eax,edx
call iprintfLF
```

Рисунок 10 — Исходный код программы variant



```
infer@Cameron: ~/lab6
infer@Cameron:~/lab6$ ./variant
Введите № студенческого билета:
1032235016
Ваш вариант: 17
infer@Cameron:~/lab6$
```

Рисунок 11 — Результат выполнения программы variant

Как видно из рисунка 11, моему номеру зачетной книжки соответствует 17 вариант.

Для вывода на экран строки «Ваш вариант» используются команды `mov eax,rem` и `call sprintf`.

Инструкции `mov ecx,x; mov edx,80;call sread` используются для считывания символов из консоли в буфер `x`.

Инструкция `call atoi` используется для преобразования символов из буфера в числовой вид.

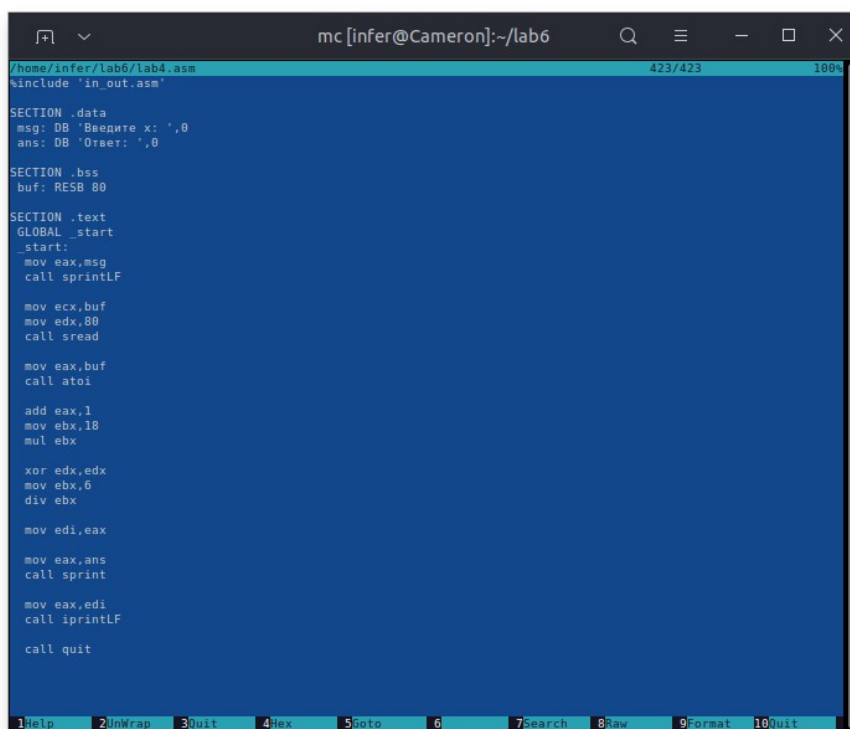
Непосредственно вычисление варианта выполняется инструкциями: `mov ebx,20;div ebx;inc edx`.

Остаток от деления записывается в регистр `edx`.

Инструкция `inc edx` используется для увеличения на единицу значения, находящегося в регистре `edx`.

За вывод результата вычисления отвечают следующие инструкции: `mov eax,edx; call iprintLF..`

Далее, согласно полученному номеру варианта создадим программу вычисляющего выражение $18 \cdot (x+1)/6$ и проверим результат работы взяв `x` равным 3 и 1. Исходный код программы показан на рисунке 12, результат выполнения на рисунке 13.



```
mc [infer@Cameron]:~/lab6 423/423 100%
/home/infer/lab6/lab4.asm
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите x: ',0
ans: DB 'Ответ: ',0

SECTION .bss
buf: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprintLF

mov ecx,buf
mov edx,80
call sread

mov eax,buf
call atoi

add eax,1
mov ebx,18
mul ebx

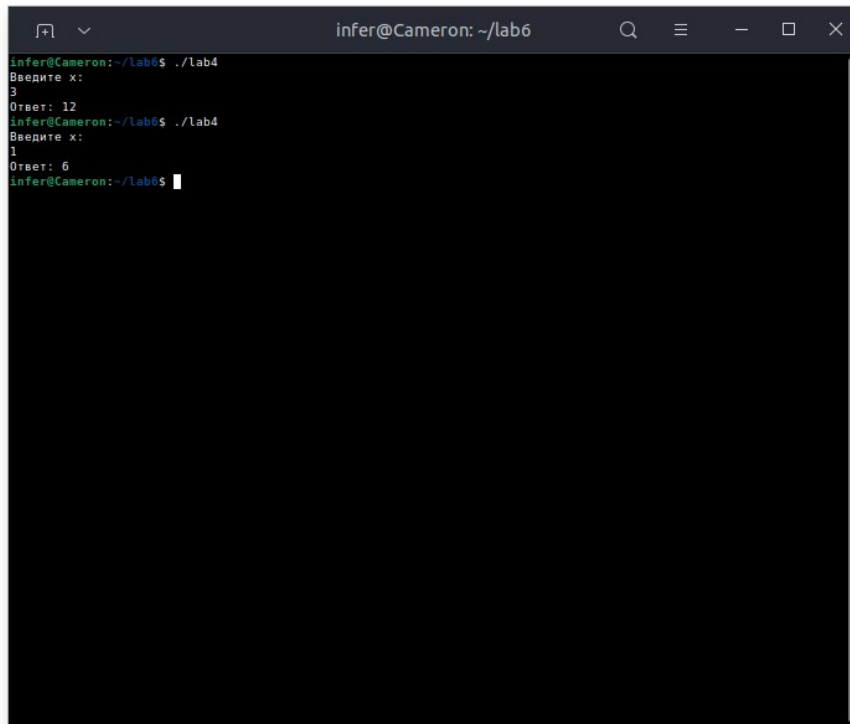
xor edx,edx
mov ebx,6
div ebx

mov edi,eax
mov eax,ans
call sprint

mov eax,edi
call iprintLF

call quit
```

Рисунок 12 — Исходный код программы lab4



```
infer@Cameron: ~/lab6
infer@Cameron:~/lab6$ ./lab4
Введите x:
3
Ответ: 12
infer@Cameron:~/lab6$ ./lab4
Введите x:
1
Ответ: 6
infer@Cameron:~/lab6$
```

Рисунок 13 — Результат выполнения программы lab4

Как несложно убедиться, выведенные ответы являются верными.

Вывод: В ходе выполнения лабораторной работы я научилась выполнять арифметические операции в NASM.