

Отчёта по лабораторной работе 7

Освоение арифметических инструкций языка ассемблера NASM.

Мягмар Уржиндорж

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	21

Список иллюстраций

2.1	Пример программы	7
2.2	Работа программы	8
2.3	Пример программы	9
2.4	Работа программы	9
2.5	Пример программы	10
2.6	Работа программы	10
2.7	Пример программы	11
2.8	Работа программы	12
2.9	Работа программы	12
2.10	Пример программы	13
2.11	Работа программы	14
2.12	Пример программы	15
2.13	Работа программы	16
2.14	Пример программы	17
2.15	Работа программы	17
2.16	Пример программы	19
2.17	Работа программы	20

Список таблиц

1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

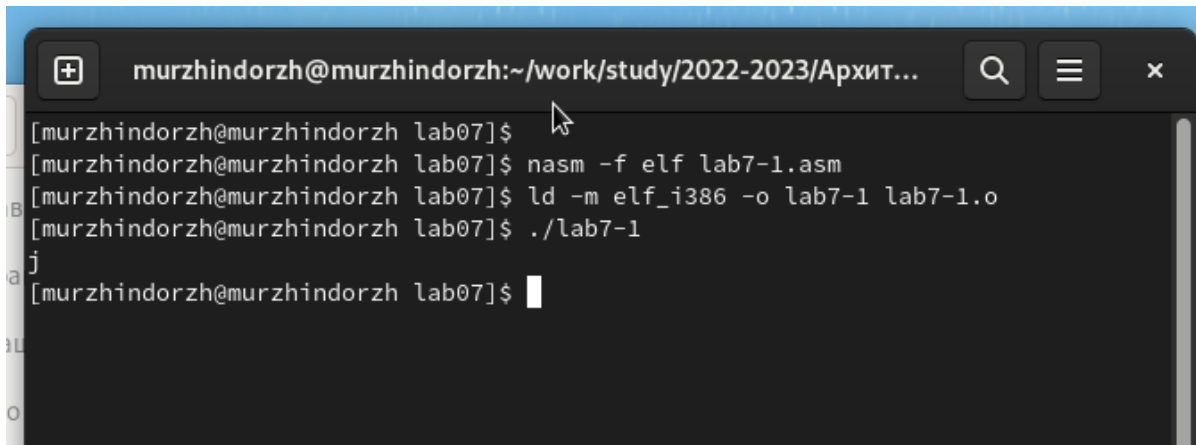
2 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab7-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax. (рис. [2.1], [2.2])



```
mc [murzhindorzh@murzhindorzh]:~/work/study/2022-  
lab7-1.asm [----] 11 L:[ 1+ 6 7/ 14] *(96 /  
%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
```

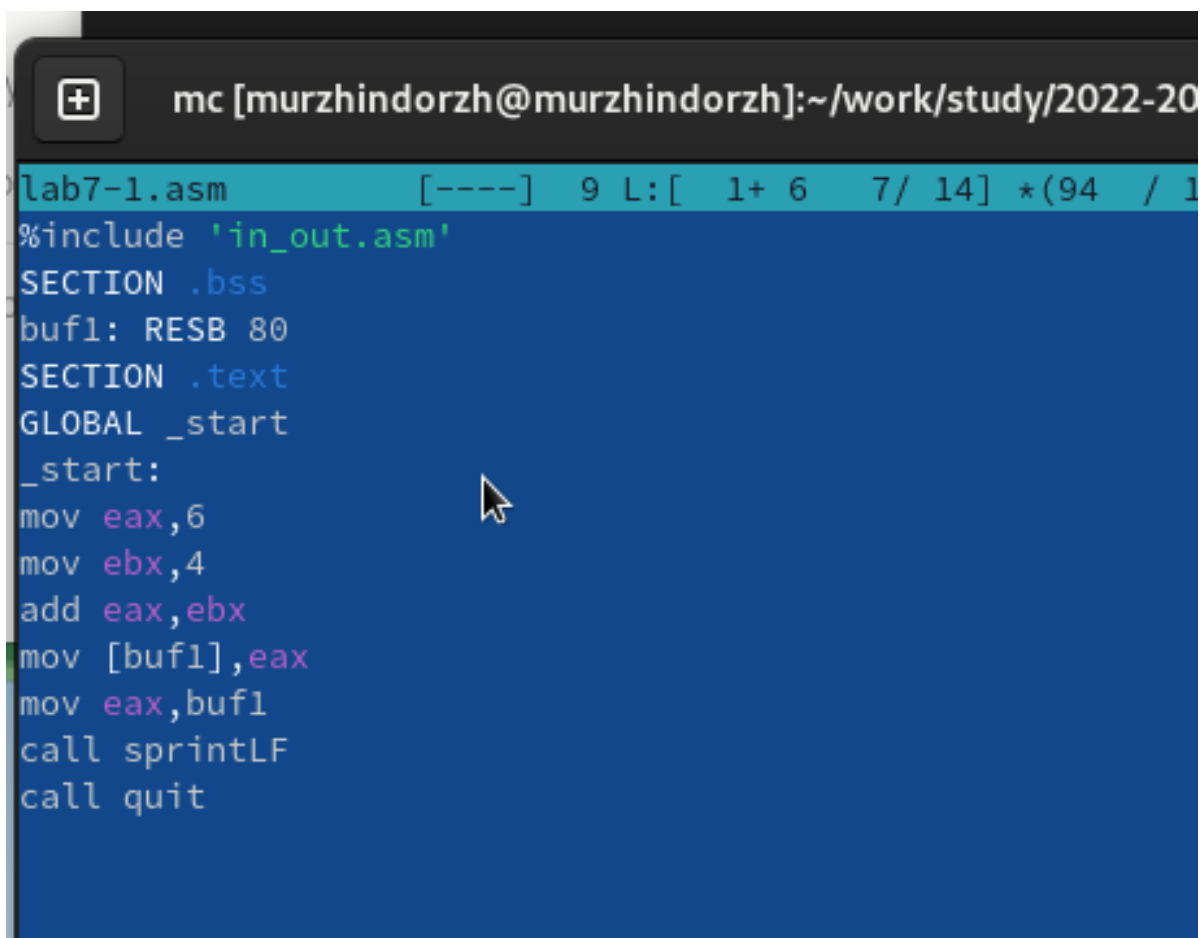
Рис. 2.1: Пример программы

A terminal window with a dark background and light text. The title bar shows the user 'murzhindorzh' and the directory '~/work/study/2022-2023/Архит...'. The terminal contains the following commands and output:

```
[murzhindorzh@murzhindorzh lab07]$  
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-1.asm  
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o  
[murzhindorzh@murzhindorzh lab07]$ ./lab7-1  
j  
[murzhindorzh@murzhindorzh lab07]$
```

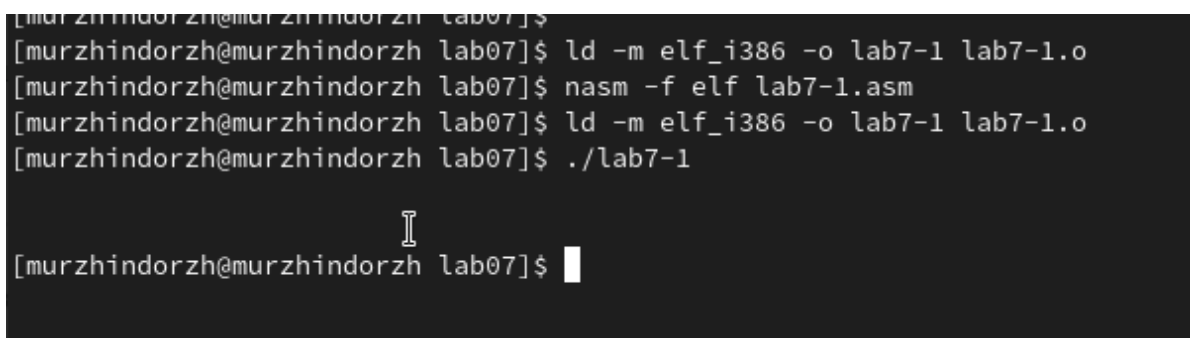
Рис. 2.2: Работа программы

3. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 1) следующим образом: (рис. [2.3], [2.4])



```
mc [murzhindorzh@murzhindorzh]:~/work/study/2022-20
lab7-1.asm [----] 9 L: [ 1+ 6 7/ 14] *(94 / 1
#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 printf
call _exit
```

Рис. 2.3: Пример программы



```
[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-1.asm
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[murzhindorzh@murzhindorzh lab07]$ ./lab7-1

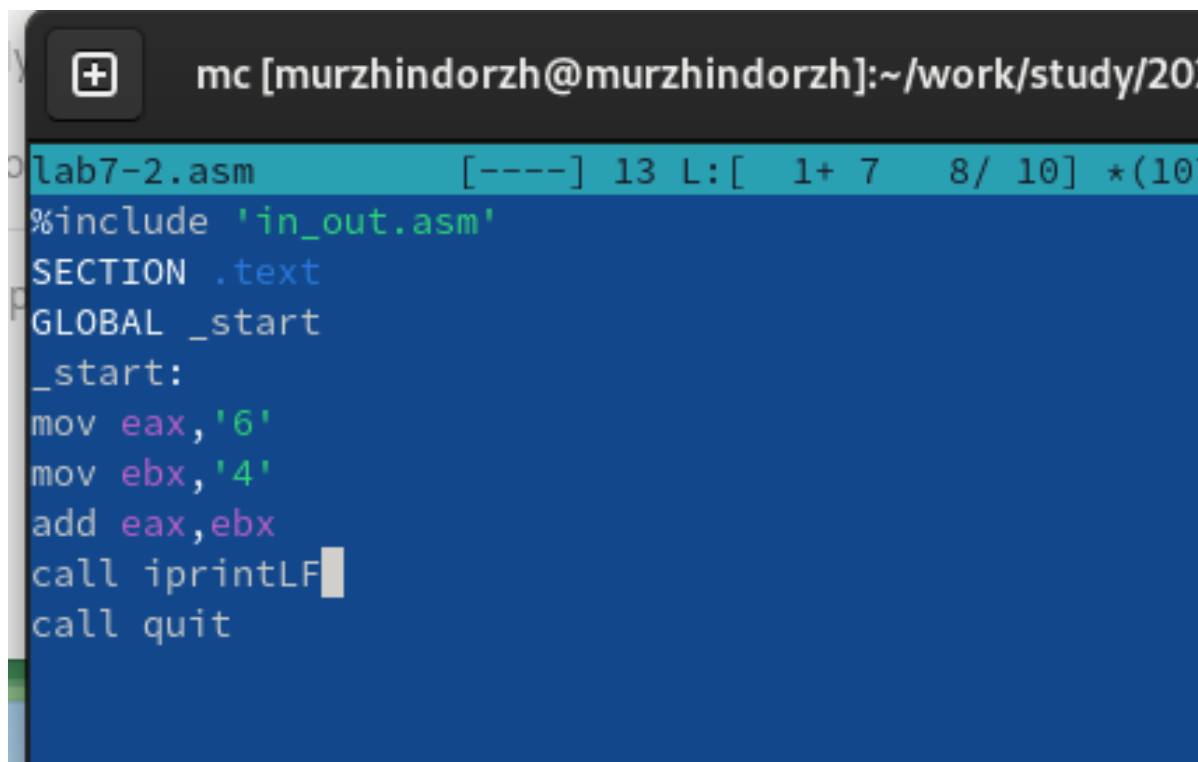
[murzhindorzh@murzhindorzh lab07]$
```

Рис. 2.4: Работа программы

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

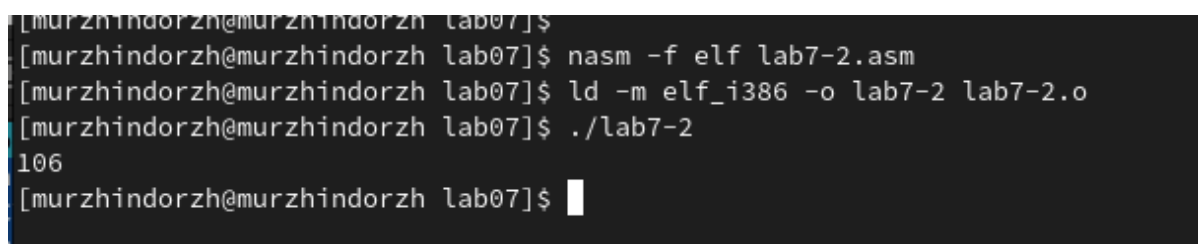
4. Как отмечалось выше, для работы с числами в файле in_out.asm реализованы

подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 7.1 с использованием этих функций. (рис. [2.5], [2.6])



```
mc [murzhindorzh@murzhindorzh]:~/work/study/201
lab7-2.asm [----] 13 L:[ 1+ 7 8/ 10] *(10
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 2.5: Пример программы



```
[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-2.asm
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[murzhindorzh@murzhindorzh lab07]$ ./lab7-2
106
[murzhindorzh@murzhindorzh lab07]$
```

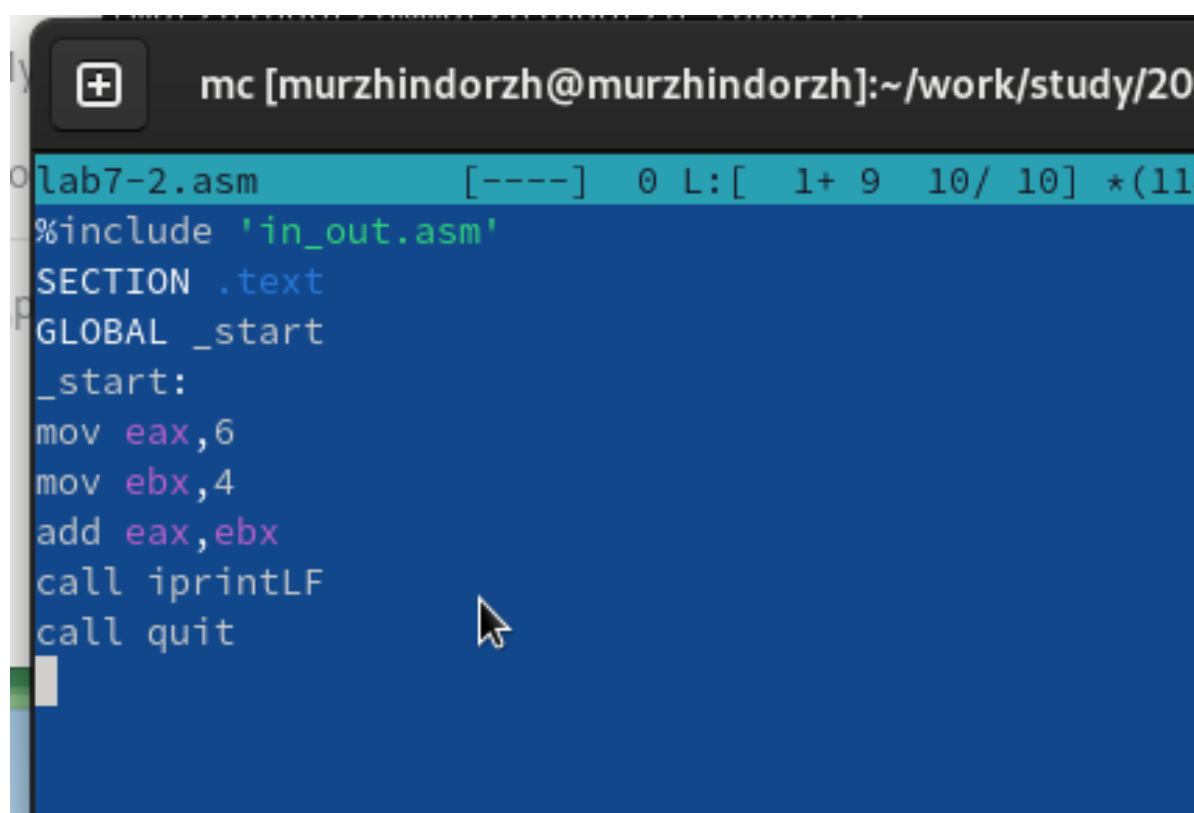
Рис. 2.6: Работа программы

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. [2.7], [2.8])

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



```
mc [murzhindorzh@murzhindorzh]:~/work/study/202
lab7-2.asm [----] 0 L:[ 1+ 9 10/ 10] *(114
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 2.7: Пример программы

```

[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-2.asm
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[murzhindorzh@murzhindorzh lab07]$ ./lab7-2
106
[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-2.asm
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[murzhindorzh@murzhindorzh lab07]$ ./lab7-2
10
[murzhindorzh@murzhindorzh lab07]$

```

Рис. 2.8: Работа программы

Замените функцию `iprintLF` на `iprint`. Создайте исполняемый файл и запустите его. Чем отличается вывод функций `iprintLF` и `iprint`? - Вывод отличается что нет переноса строки. (рис. [2.9])

```

[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-2.asm
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[murzhindorzh@murzhindorzh lab07]$ ./lab7-2
10
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-2.asm
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[murzhindorzh@murzhindorzh lab07]$ ./lab7-2
10[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$

```

Рис. 2.9: Работа программы

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

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

. (рис. [2.10], рис. [2.11])



```
mc [murzhindorzh@murzhindorzh]
lab7-3.asm [-] 9 L: [ 1+1
#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. 2.10: Пример программы

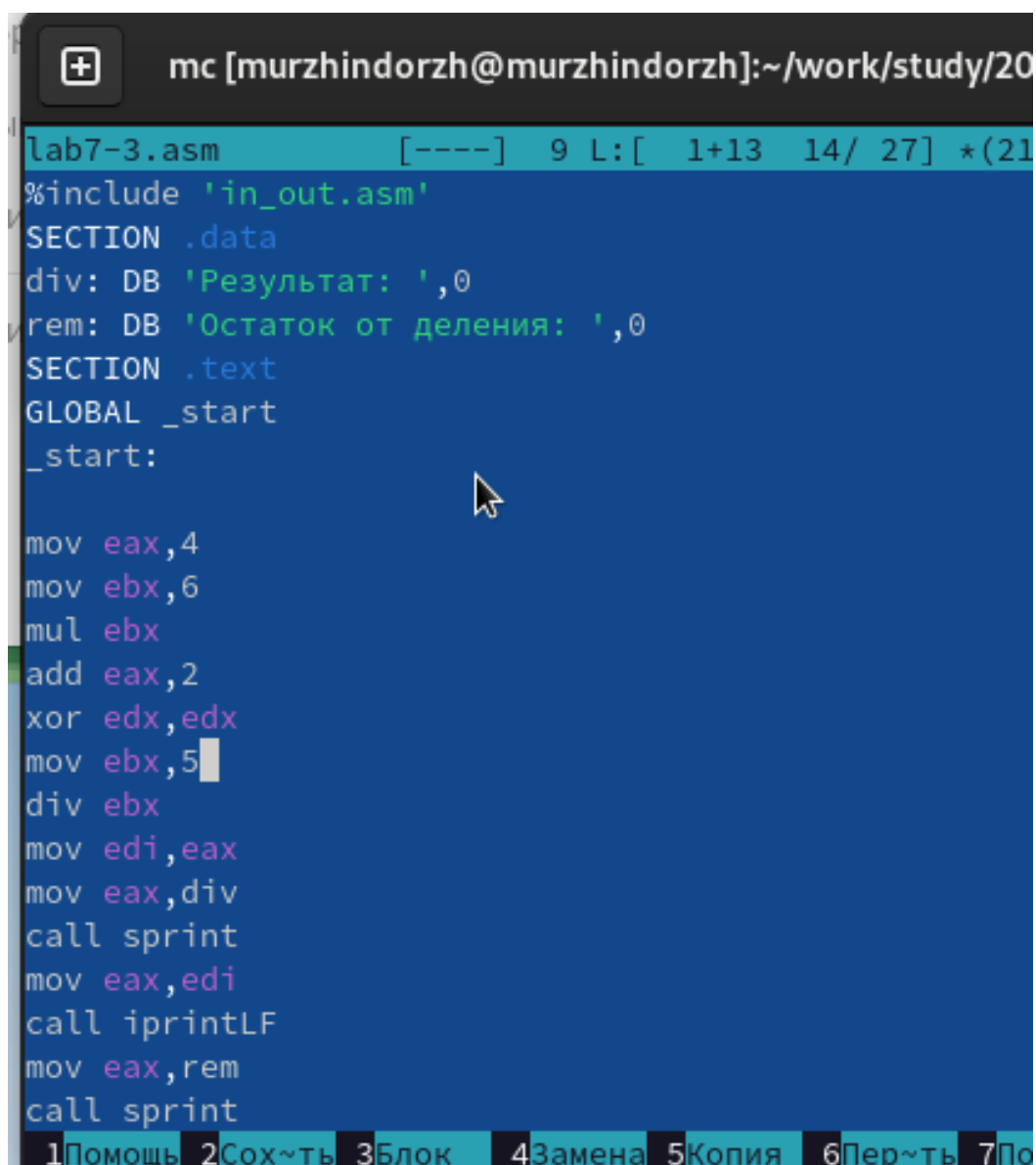
```
[murzhindorzh@murzhindorzh lab07]$  
[murzhindorzh@murzhindorzh lab07]$  
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-3.asm  
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o  
[murzhindorzh@murzhindorzh lab07]$ ./lab7-3  
Результат: 4  
Остаток от деления: 1  
[murzhindorzh@murzhindorzh lab07]$
```

Рис. 2.11: Работа программы

Измените текст программы для вычисления выражения

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

. Создайте исполняемый файл и проверьте его работу. (рис. [2.12], рис. [2.13])



```
mc [murzhindorzh@murzhindorzh]:~/work/study/20
lab7-3.asm [----] 9 L: [ 1+13 14/ 27] *(21
#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
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск

Рис. 2.12: Пример программы

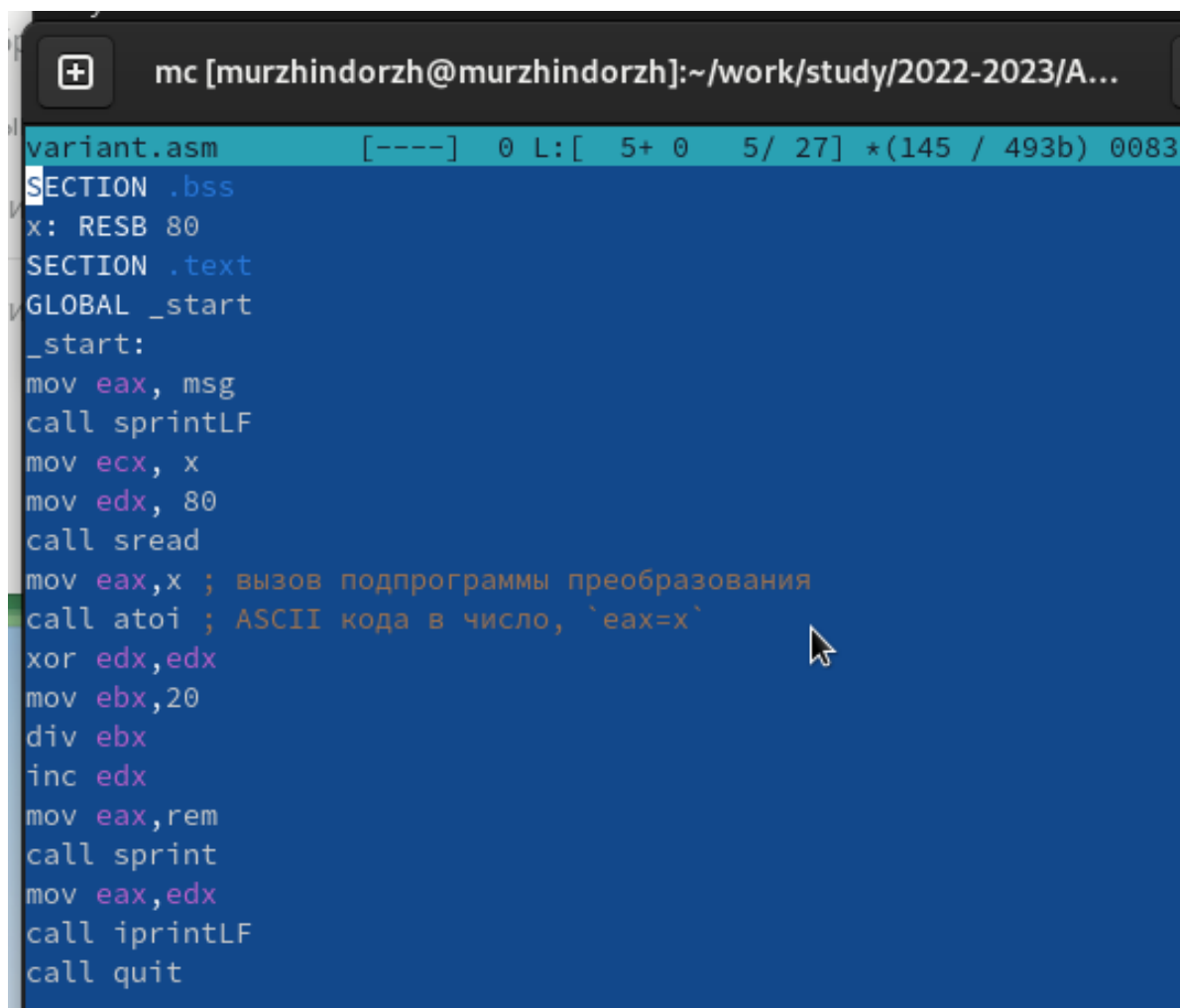
```

4 [murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-3.asm
3 [murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[murzhindorzh@murzhindorzh lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf lab7-3.asm
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[murzhindorzh@murzhindorzh lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[murzhindorzh@murzhindorzh lab07]$ █

```


Рис. 2.13: Работа программы

7. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму: (рис. [2.14], рис. [2.15])



```
mc [murzhindorzh@murzhindorzh]:~/work/study/2022-2023/A...
variant.asm [----] 0 L: [ 5+ 0 5/ 27] *(145 / 493b) 0083
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. 2.14: Пример программы



```
[murzhindorzh@murzhindorzh lab07]$
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf variant.asm
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o variant variant.o
[murzhindorzh@murzhindorzh lab07]$ ./variant
Введите № студенческого билета:
1032228006
Ваш вариант: 7
[murzhindorzh@murzhindorzh lab07]$
```

Рис. 2.15: 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? `push ecx, x` `mov edx, 80` `call sread`

Считывает значение студбилета в переменную X из консоли

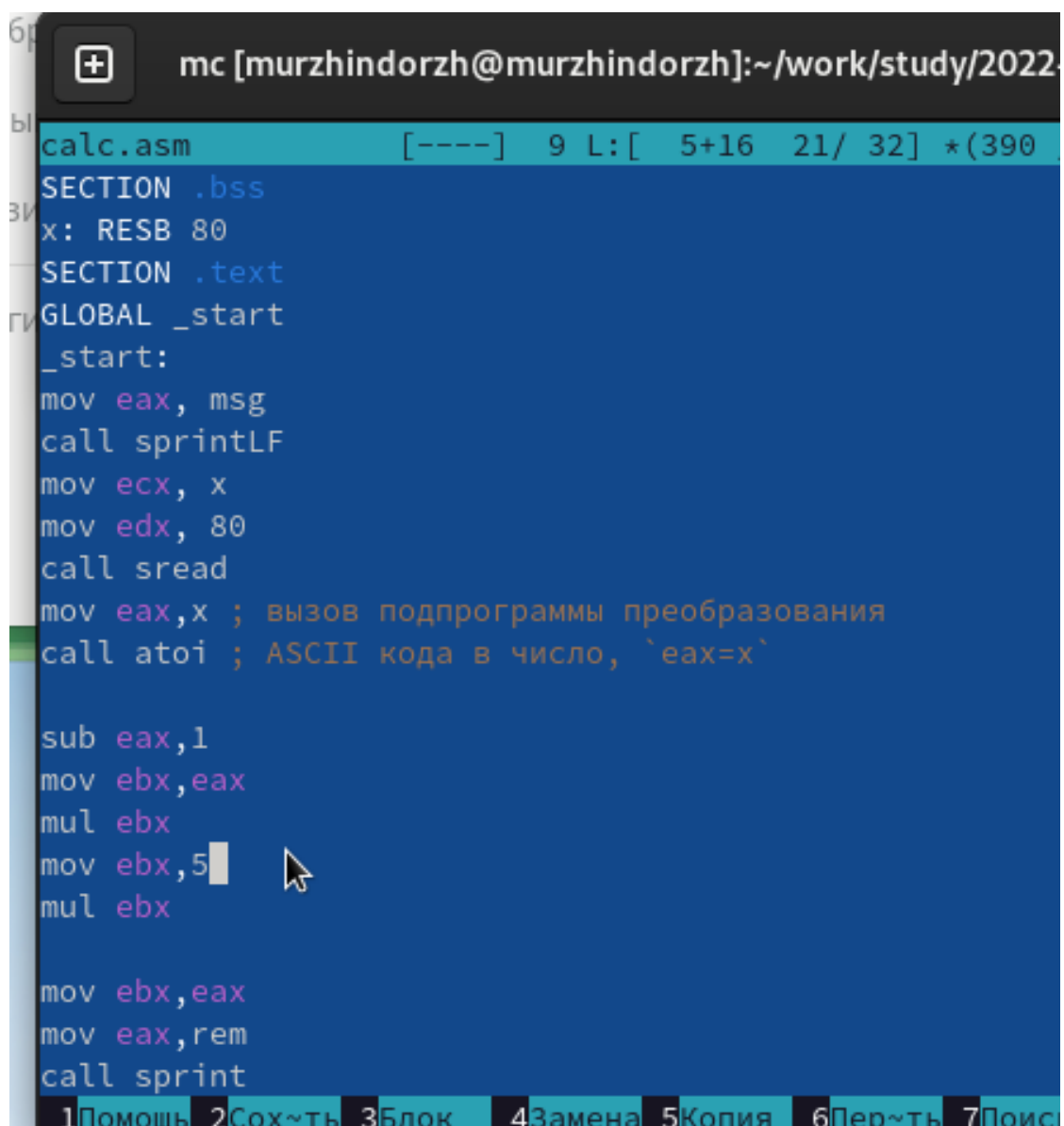
- Для чего используется инструкция “`call atoi`”? - эта подпрограмма переводит введенные символы в числовой формат
- Какие строки листинга 7.4 отвечают за вычисления варианта? `xor edx,edx` `mov ebx,20` `div ebx`
- В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”? 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 вычислений `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. [2.16], рис. [2.17])

Получили вариант 7 -

$$5(x - 1)^2$$

для $x=3$ и 5



```
mc [murzhindorzh@murzhindorzh]:~/work/study/2022
calc.asm [-----] 9 L: [ 5+16 21/ 32] *(390
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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`

sub eax, 1
mov ebx, eax
mul ebx
mov ebx, 5
mul ebx

mov ebx, eax
mov eax, rem
call sprint
1Помощь 2Сох~ть 3Блок 4Замена 5Копия 6Пер~ть 7Поиск
```

Рис. 2.16: Пример программы

```
[murzhindorzh@murzhindorzh lab07]$  
[murzhindorzh@murzhindorzh lab07]$  
[murzhindorzh@murzhindorzh lab07]$ nasm -f elf calc.asm  
[murzhindorzh@murzhindorzh lab07]$ ld -m elf_i386 -o calc calc.o  
[murzhindorzh@murzhindorzh lab07]$ ./calc  
Введите X  
3  
выражение = : 20  
[murzhindorzh@murzhindorzh lab07]$ ./calc  
Введите X  
5  
выражение = : 80  
[murzhindorzh@murzhindorzh lab07]$
```

Рис. 2.17: Работа программы

3 Выводы

Изучили работу с арифметическими операциями