

# **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**

**Дисциплина: Архитектура компьютера**

Обрезкова Анастасия Владимировна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Выполнение арифметических операций в NASM. . . . .	12
4.2	Ответы на вопросы . . . . .	15
4.3	Задания для самостоятельной работы . . . . .	17
<b>5</b>	<b>Выводы</b>	<b>20</b>
	<b>Список литературы</b>	<b>21</b>

## Список иллюстраций

4.1	Создание, переход в lab07 . . . . .	8
4.2	Ввела нужный текст . . . . .	9
4.3	Вывела результат . . . . .	9
4.4	Изменения текста . . . . .	10
4.5	Результат изменений . . . . .	10
4.6	Создание файла . . . . .	10
4.7	Ввела текст программы . . . . .	11
4.8	Результат программы . . . . .	11
4.9	Ввела изменения в текст . . . . .	11
4.10	Результат изменений . . . . .	12
4.11	Замена функции . . . . .	12
4.12	Вывод результата . . . . .	12
4.13	Введение текста . . . . .	13
4.14	Результат программы . . . . .	13
4.15	Изменение текста . . . . .	14
4.16	Результат вычислений . . . . .	14
4.17	Введенный текст . . . . .	15
4.18	Результат программы . . . . .	15
4.19	Текст программы . . . . .	18
4.20	Текст программы . . . . .	19
4.21	Результат . . . . .	19

## Список таблиц

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Задание

Освоить арифметические инструкции языка ассемблера NASM.

### 3 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти.

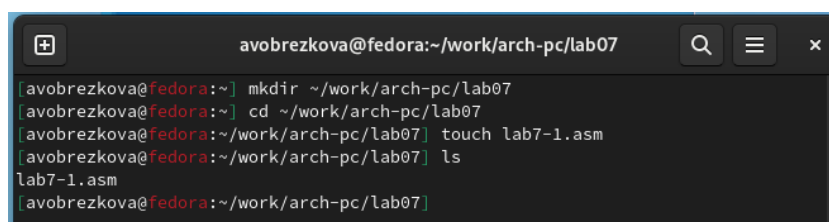
ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно стандарту ASCII каждый символ кодируется одним байтом.

Для выполнения лабораторных работ в файле `in_out.asm` реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Это:

- `iprint` – вывод на экран чисел в формате ASCII, перед вызовом `iprint` в регистр `eax` необходимо записать выводимое число (`mov eax,` ).
- `iprintLF` – работает аналогично `iprint`, но при выводе на экран после числа добавляет к символ перевода строки.
- `atoi` – функция преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`, перед вызовом `atoi` в регистр `eax` необходимо записать число (`mov eax,` ).

## 4 Выполнение лабораторной работы

1. Создала каталог для программ лабораторной работы №7, перешла в него и создала файл lab7-1.asm. (рис. 4.1)

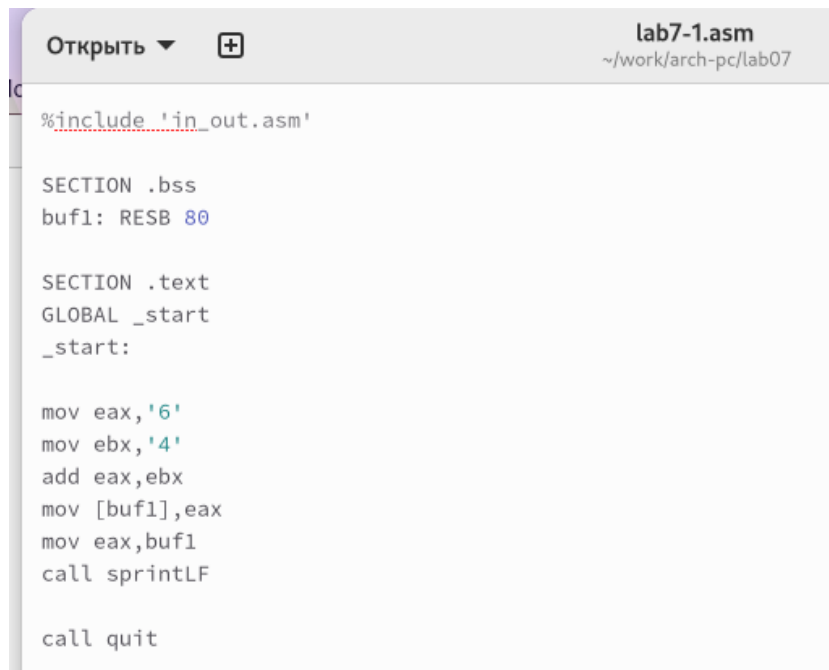


```
avobrezkova@fedora:~/work/arch-pc/lab07
[avobrezkova@fedora:~] mkdir ~/work/arch-pc/lab07
[avobrezkova@fedora:~] cd ~/work/arch-pc/lab07
[avobrezkova@fedora:~/work/arch-pc/lab07] touch lab7-1.asm
[avobrezkova@fedora:~/work/arch-pc/lab07] ls
lab7-1.asm
[avobrezkova@fedora:~/work/arch-pc/lab07]
```

Рис. 4.1: Создание, переход в lab07

2. Ввела в файл lab7-1 нужный текст программы из листинга 7.1., создала исполняемый файл и вывела результат. (рис. 4.2; рис. 4.3)





The screenshot shows a text editor window titled "lab7-1.asm" with the path "~/work/arch-pc/lab07". The code inside is as follows:

```
Открыть + lab7-1.asm
~/work/arch-pc/lab07

%include 'in_out.asm'

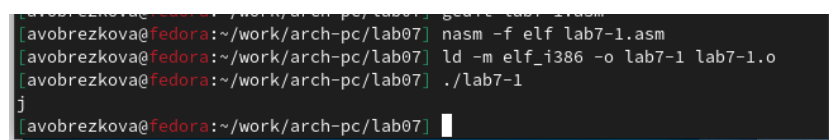
SECTION .bss
buf1: RESB 80

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. 4.2: Ввела нужный текст

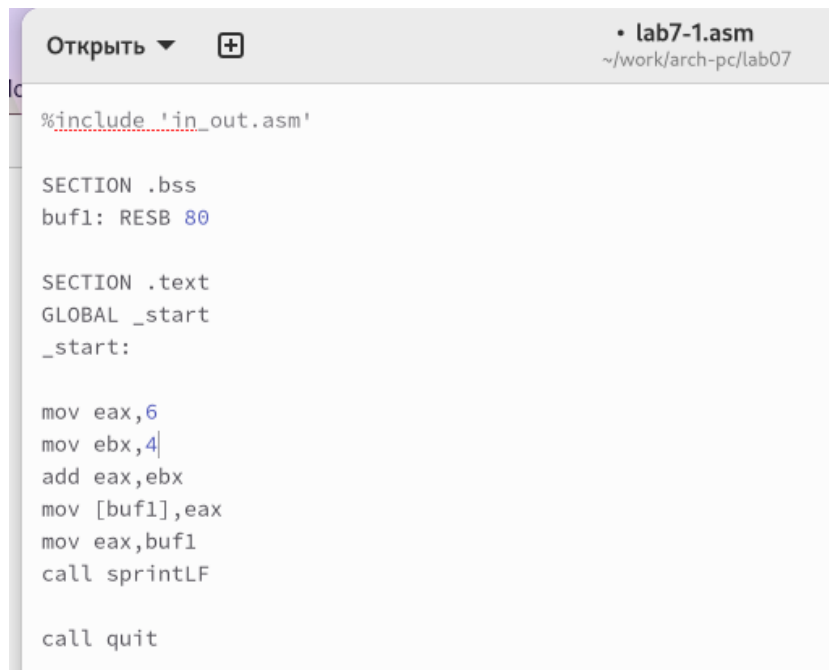


The screenshot shows a terminal window with the following commands and output:

```
avobrezkova@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
avobrezkova@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
avobrezkova@fedora:~/work/arch-pc/lab07$ ./lab7-1
6
j
avobrezkova@fedora:~/work/arch-pc/lab07$
```

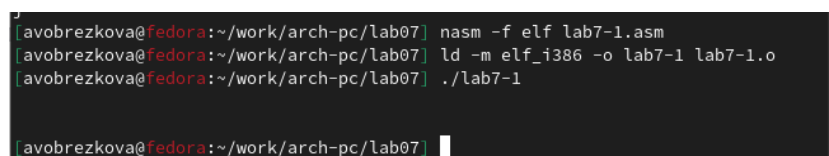
Рис. 4.3: Вывела результат

3. Изменила текст программы и вместо символов, записала в регистры числа.  
(рис. 4.4; рис. 4.5)

A screenshot of a text editor window titled 'lab7-1.asm' with the path '~/.work/arch-pc/lab07'. The code includes a directive to include 'in\_out.asm', defines a .bss section with a buffer 'buf1' of 80 bytes, and a .text section starting at '\_start'. The assembly code moves the value 6 into 'eax', 4 into 'ebx', adds them, stores the result in 'buf1', prints it with 'sprintLF', and finally calls 'quit'.

```
Открыть ▾ + • lab7-1.asm  
~/.work/arch-pc/lab07  
  
%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. 4.4: Изменения текста

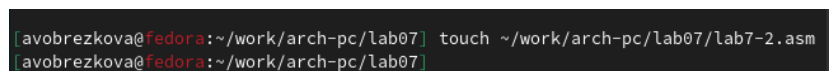
A screenshot of a terminal window showing the compilation of 'lab7-1.asm'. The user runs 'nasm -f elf lab7-1.asm', then 'ld -m elf\_i386 -o lab7-1 lab7-1.o', and finally './lab7-1'. The prompt shows the user is 'avobrezkova' on a 'fedora' machine in the directory '~/.work/arch-pc/lab07'.

```
[avobrezkova@fedora:~/work/arch-pc/lab07] nasm -f elf lab7-1.asm  
[avobrezkova@fedora:~/work/arch-pc/lab07] ld -m elf_i386 -o lab7-1 lab7-1.o  
[avobrezkova@fedora:~/work/arch-pc/lab07] ./lab7-1  
  
[avobrezkova@fedora:~/work/arch-pc/lab07] █
```

Рис. 4.5: Результат изменений

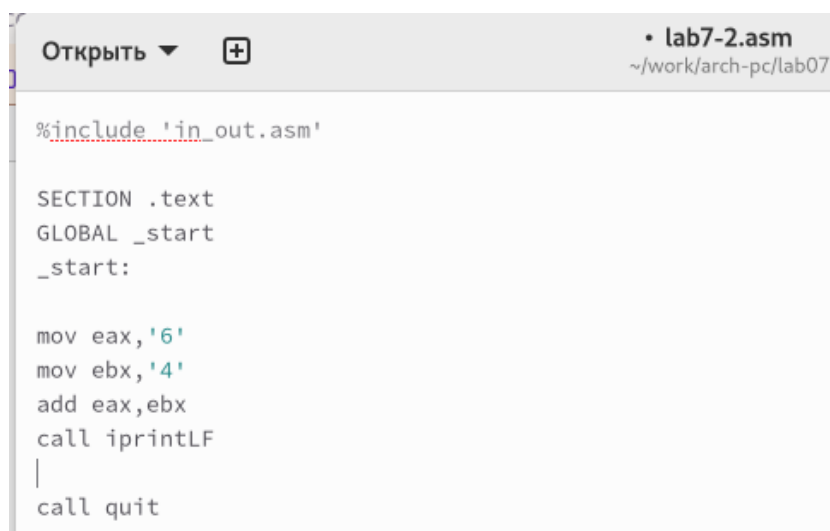
Этот символ не отображается при выводе на экран.

4. Создала файл lab7-2.asm в нужном каталоге, ввела в него текст программы из листинга 7.2. и вывела результат. (рис. 4.6; рис. 4.7; рис. 4.8)

A screenshot of a terminal window showing the creation of a new file 'lab7-2.asm' using the 'touch' command. The prompt shows the user is 'avobrezkova' on a 'fedora' machine in the directory '~/.work/arch-pc/lab07'.

```
[avobrezkova@fedora:~/work/arch-pc/lab07] touch ~/.work/arch-pc/lab07/lab7-2.asm  
[avobrezkova@fedora:~/work/arch-pc/lab07]
```

Рис. 4.6: Создание файла



The screenshot shows a text editor window titled "lab7-2.asm" with the path "~/work/arch-pc/lab07". The code inside is as follows:

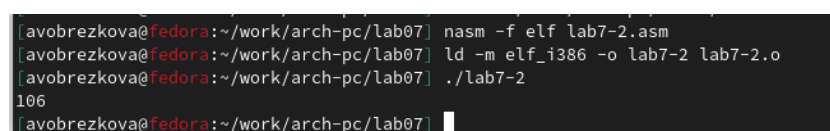
```
Открыть ▾ + lab7-2.asm
~/work/arch-pc/lab07

%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
|
call quit
```

Рис. 4.7: Ввела текст программы

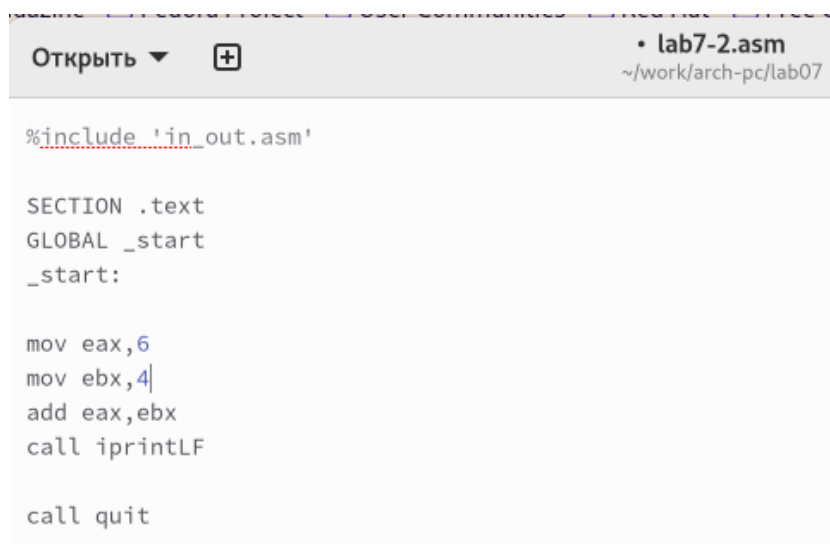


The screenshot shows a terminal window with the following commands and output:

```
avobrezkova@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
avobrezkova@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
avobrezkova@fedora:~/work/arch-pc/lab07$ ./lab7-2
106
```

Рис. 4.8: Результат программы

5. Изменила символы на числа, создала исполняемый файл и вывела результат.  
(рис. 4.9; рис. 4.10)



The screenshot shows the same text editor window as in Figure 4.7, but with the assembly code modified to use numbers instead of characters:

```
Открыть ▾ + lab7-2.asm
~/work/arch-pc/lab07

%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 4.9: Ввела изменения в текст

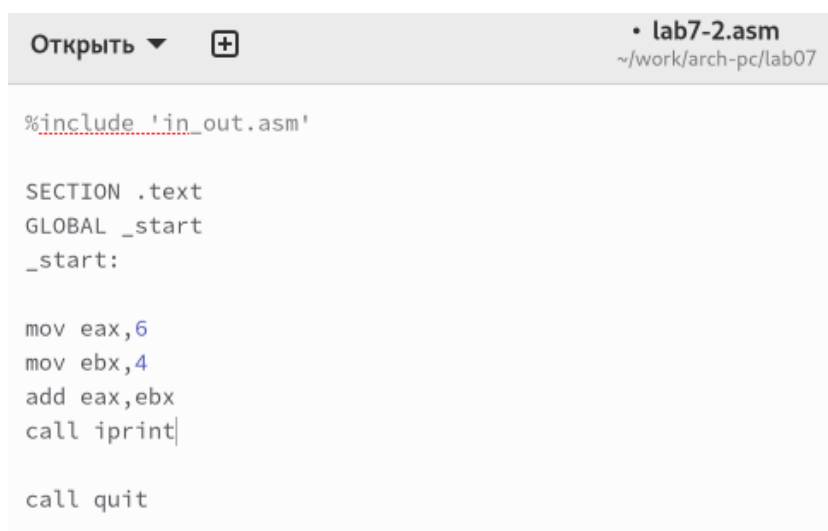
```

[avobrezkova@fedora:~/work/arch-pc/lab07] nasm -f elf lab7-2.asm
[avobrezkova@fedora:~/work/arch-pc/lab07] ld -m elf_i386 -o lab7-2 lab7-2.o
[avobrezkova@fedora:~/work/arch-pc/lab07] ./lab7-2
10
[avobrezkova@fedora:~/work/arch-pc/lab07]

```

Рис. 4.10: Результат изменений

6. Заменяла функцию `iprintLF` на `iprint` и вывела результат. (рис. 4.11; рис. 4.12)



```

Открыть ▾ + lab7-2.asm
~/work/arch-pc/lab07

%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

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

call quit

```

Рис. 4.11: Замена функции

```

10
[avobrezkova@fedora:~/work/arch-pc/lab07] nasm -f elf lab7-2.asm
[avobrezkova@fedora:~/work/arch-pc/lab07] ld -m elf_i386 -o lab7-2 lab7-2.o
[avobrezkova@fedora:~/work/arch-pc/lab07] ./lab7-2
10
[avobrezkova@fedora:~/work/arch-pc/lab07]

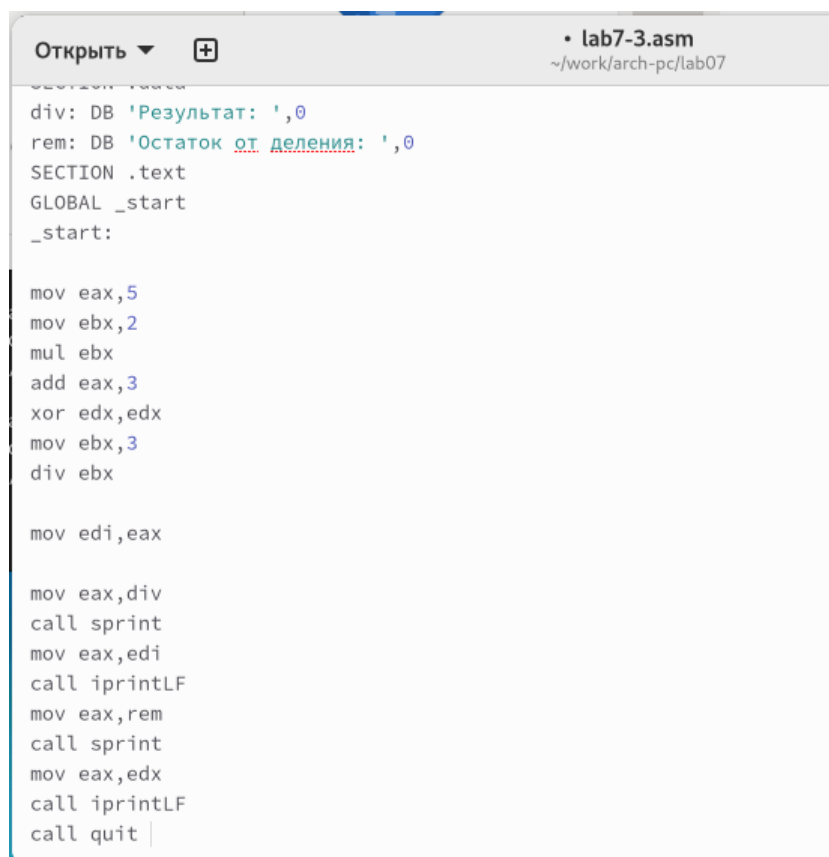
```

Рис. 4.12: Вывод результата

Вывод функции `iprintLF` и `iprint` отличаются выводом результата, в одном варианте результат выводится на отдельной строчке, а в другом нет.

## 4.1 Выполнение арифметических операций в NASM.

1. Создала файл `lab7-3.asm` в нужном каталоге, ввела нужный текст и вывела результат. (рис. 4.13; рис. 4.14)



```
Открыть ▾ + lab7-3.asm
~/work/arch-pc/lab07

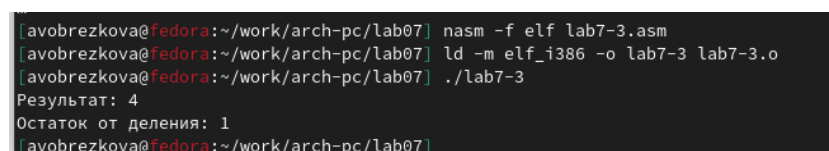
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:

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 sprintf
mov eax,edi
call iprintLF
mov eax,rem
call sprintf
mov eax,edx
call iprintLF
call quit |
```

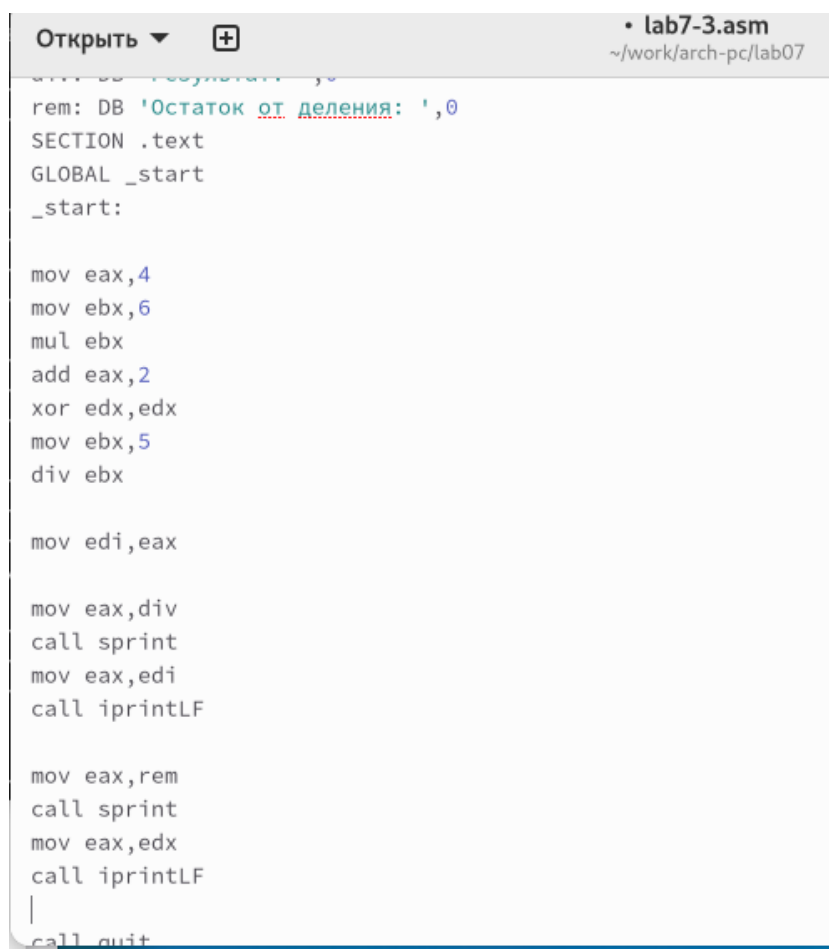
Рис. 4.13: Введение текста



```
avobrezkova@fedora:~/work/arch-pc/lab07 nasm -f elf lab7-3.asm
avobrezkova@fedora:~/work/arch-pc/lab07 ld -m elf_i386 -o lab7-3 lab7-3.o
avobrezkova@fedora:~/work/arch-pc/lab07 ./lab7-3
Результат: 4
Остаток от деления: 1
avobrezkova@fedora:~/work/arch-pc/lab07
```

Рис. 4.14: Результат программы

2. Изменила текст программы для вычисления выражения  $f(x)=(4*6+2)/5$ . (рис. 4.15; рис. 4.16)



```
Открыть ▾ + lab7-3.asm
~/work/arch-pc/lab07
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:

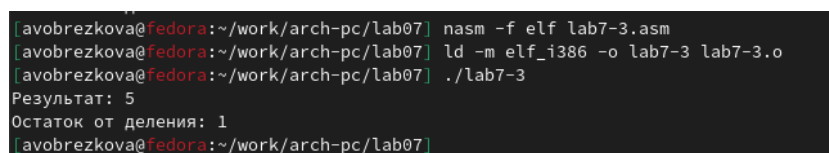
mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

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
```

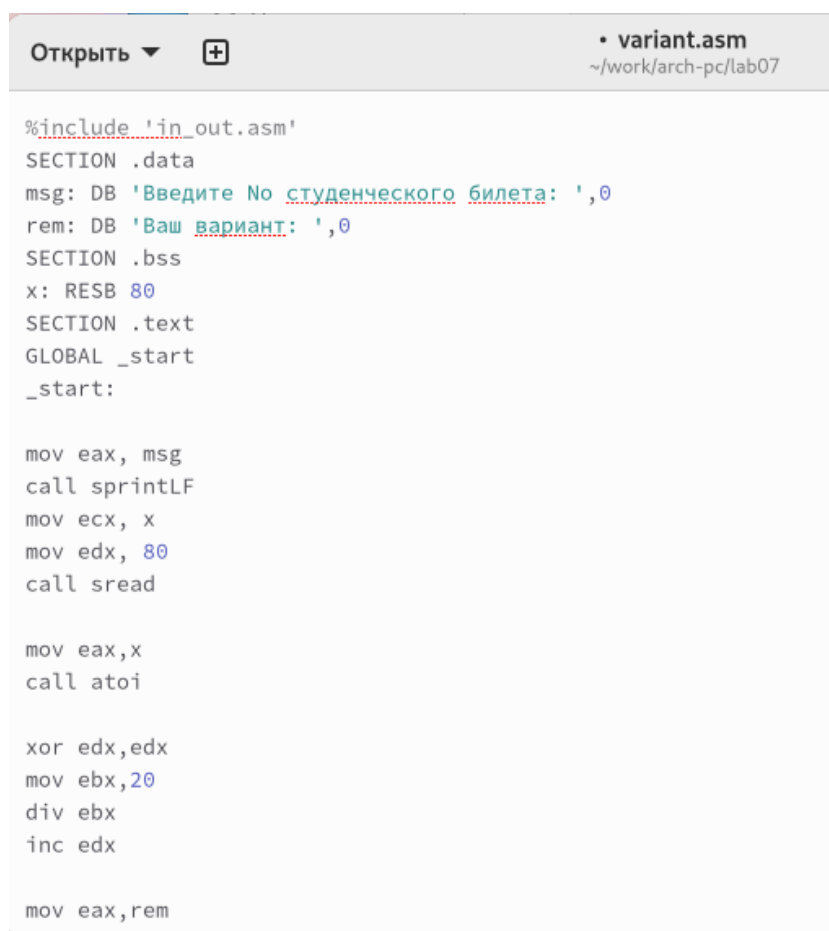
Рис. 4.15: Изменение текста



```
[avobrezkova@fedora:~/work/arch-pc/lab07] nasm -f elf lab7-3.asm
[avobrezkova@fedora:~/work/arch-pc/lab07] ld -m elf_i386 -o lab7-3 lab7-3.o
[avobrezkova@fedora:~/work/arch-pc/lab07] ./lab7-3
Результат: 5
Остаток от деления: 1
[avobrezkova@fedora:~/work/arch-pc/lab07]
```

Рис. 4.16: Результат вычислений

3. Изучила текст программы из листинга 7.4., ввела его в нужный файл и вывела результат. (рис. 4.17; рис. 4.18)



```
Открыть ▾ + • variant.asm
~/work/arch-pc/lab07

%include 'in_out.asm'
SECTION .data
msg: DB 'Введите No студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:

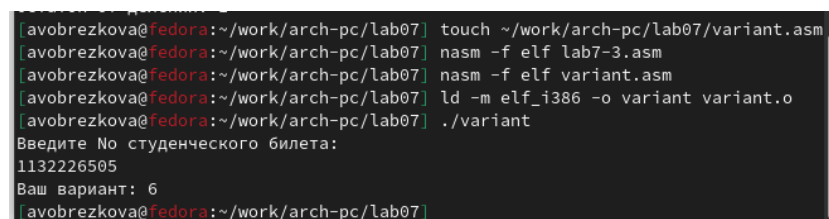
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

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

mov eax, rem
```

Рис. 4.17: Введенный текст



```
[avobrezkova@fedora:~/work/arch-pc/lab07] touch ~/work/arch-pc/lab07/variant.asm
[avobrezkova@fedora:~/work/arch-pc/lab07] nasm -f elf lab7-3.asm
[avobrezkova@fedora:~/work/arch-pc/lab07] nasm -f elf variant.asm
[avobrezkova@fedora:~/work/arch-pc/lab07] ld -m elf_i386 -o variant variant.o
[avobrezkova@fedora:~/work/arch-pc/lab07] ./variant
Введите No студенческого билета:
1132226505
Ваш вариант: 6
[avobrezkova@fedora:~/work/arch-pc/lab07]
```

Рис. 4.18: Результат программы

## 4.2 Ответы на вопросы

1. Какие строки листинга 7.4. отвечают за вывод на экран сообщения 'Ваш вариант:'?

```
mov eax, msg call sprintfLF
```

2. Для чего используются следующие инструкции? `mov ecx, x` `mov edx, 80`  
`call sread`

Эти инструкции используются для ввода переменной X с клавиатуры и сохранения введенных данных.

3. Для чего используется инструкция “`call atoi`”?

Эта инструкция используется для преобразования кода переменной ASCII в число.

4. Какие строки листинга 7.4. отвечают за вычисления варианта?

```
mov ebx, 20 div ebx inc edx
```

5. В какой регистр записывается остаток от деления при выполнении инструкции “`div edx`”?

В регистре `ebx`.

6. Для чего используется инструкция “`inc edx`”?

Для увеличения значения `edx` на 1.


7. Какие строки листинга 7.4. отвечают за вывод на экран результата вычислений?

```
mov eax, edx call iprintLF
```



### 4.3 Задания для самостоятельной работы

1. Я написала программу, которая будет решать выражение и выводить ответ при введенных X. Для решения мне попалось уравнение из 6 варианта:  
 $f(x)=x^{3/2}+1$ . (рис. 4.19; рис. 4.20; рис. 4.21)

Открыть ▾ 

2.asm  
~/work/arch-pc/lab07

```
%include 'in_out.asm'

SECTION .data
rem: DB 'Вычислить значение выражения  $x^3/2+1$ ',0
msg: DB 'Введите x: ',0
div: DB 'Результат: ',0

SECTION .bss
x:     RESB 80
rez:   RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, rem
call sprintfLF
|
mov eax, msg
call sprintfLF

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

mov ecx, eax
mul ecx
mul ecx
mov ebx, 2
div ebx
add eax, 1

mov [rez], eax

mov eax, div
call sprintfLF
mov eax, [rez]
```

Рис. 4.19: Текст программы

```

mov [rez], eax

mov eax, div
call sprintLF
mov eax, [rez]
call iprintLF

call quit

```

Рис. 4.20: Текст программы

```

[avobrezkova@fedora:~/work/arch-pc/lab07] nasm -f elf 2.asm
[avobrezkova@fedora:~/work/arch-pc/lab07] ld -m elf_i386 -o 2 2.o
[avobrezkova@fedora:~/work/arch-pc/lab07] ./2
Вычислить значение выражения x^3/2+1
Введите x:
2
Результат:
5
[avobrezkova@fedora:~/work/arch-pc/lab07] ./2
Вычислить значение выражения x^3/2+1
Введите x:
5
Результат:
63
[avobrezkova@fedora:~/work/arch-pc/lab07]

```

Рис. 4.21: Результат

Данные изменения можно проверить по ссылке: [https://github.com/avobrezkova/study\\_2022-2023\\_arh-pc/tree/master/labs/lab07](https://github.com/avobrezkova/study_2022-2023_arh-pc/tree/master/labs/lab07)

## 5 Выводы

Освоила арифметические инструкции языка ассемблера NASM.

## Список литературы

1. [https://esystem.rudn.ru/pluginfile.php/1584388/mod\\_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%967.pdf](https://esystem.rudn.ru/pluginfile.php/1584388/mod_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%967.pdf)