

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Студент: Гасанова Шакира Чингизовна

Группа: НКАбд-05-24

МОСКВА

2024 г.

# Содержание

<b>1 Цель работы .....</b>	<b>3</b>
<b>2 Задание .....</b>	<b>4</b>
<b>3 Теоретическое введение .....</b>	<b>5</b>
<b>4 Выполнение лабораторной работы</b>	
4.1 Символьные и численные данные в NASM.....	6
4.2 Выполнение арифметических операций в NASM.....	10
4.2.1 Ответы на вопросы по программе.....	12
4.3 Выполнение заданий для самостоятельной работы.....	13
<b>5 Выводы .....</b>	<b>16</b>
<b>6 Источники .....</b>	<b>17</b>

## **1 Цель работы**

Целью данной лабораторной работы является освоение арифметических инструкций языка ассемблера NASM.

## **2 Задание**

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

### 3 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов, для которых необходимо указать, где находятся обрабатываемые данные. Операнд может находиться в регистре или в ячейке памяти. Существуют три основных способа адресации:

1. Регистровая адресация: данные хранятся в регистрах, и команда содержит указание на их имена. Например: `mov ax, bx`.

2. Непосредственная адресация: значение операнда записывается непосредственно в команде. Например: `mov ax, 2`.

3. Адресация памяти: команда содержит указание на адрес в памяти, где находятся данные. В этом случае используется символическое обозначение ячейки памяти, над содержимым которой выполняется операция.

Ввод данных с клавиатуры и их вывод на экран происходит в виде символов. Для кодировки символов используется таблица ASCII (American Standard Code for Information Interchange – Американский стандартный код для обмена информацией). Согласно этому стандарту, каждый символ кодируется одним байтом.

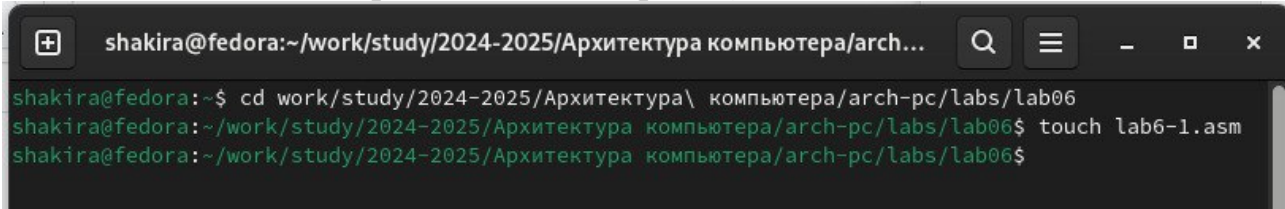
Инструкции NASM не предусматривают непосредственного вывода чисел в их числовом представлении. Поэтому, чтобы вывести число, его необходимо сначала преобразовать в ASCII-коды соответствующих цифр и выводить уже эти коды. В противном случае, при выводе числа без преобразования, оно будет интерпретировано как последовательность ASCII-символов, соответствующих байтам числа. Экран отобразит символы, а не само число.

Аналогичная ситуация возникает при вводе данных с клавиатуры. Введенные значения воспринимаются как символы, что делает невозможным выполнение арифметических операций без предварительного преобразования. Для корректной работы необходимо преобразовывать символы ASCII в числа и обратно.

## 4 Выполнение лабораторной работы

### 4.1 Символьные и численные данные в NASM

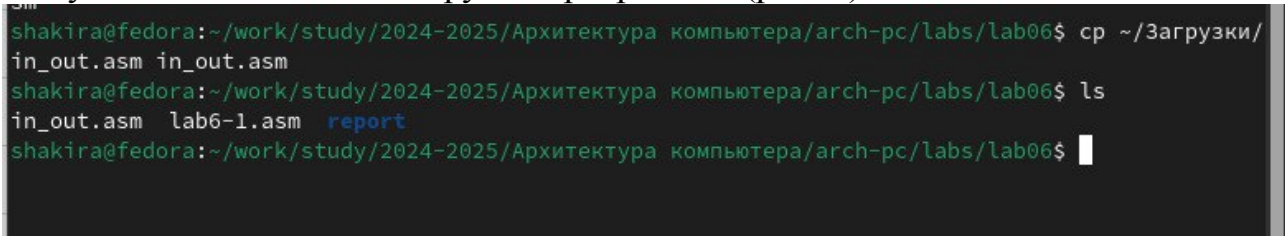
Перехожу в созданный каталог с помощью утилиты `cd` и с помощью команды `touch` создаю файл `lab6-1.asm` (рис. 1).



```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-...  
shakira@fedora:~$ cd work/study/2024-2025/Архитектура\ компьютера/arch-pc/labs/lab06  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ touch lab6-1.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.1 Создание файла

Копирую в текущий каталог файл `in_out.asm` с помощью утилиты `cp`, т.к. он будет использоваться в других программах (рис. 2).



```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ cp ~/Загрузки/  
in_out.asm in_out.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ls  
in_out.asm lab6-1.asm report  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.2 Копирование файла

Открываю созданный файл `lab6-1.asm`, вставляю в него программу вывода значения регистра `eax` (рис. 3).



```
Открыть ▾ + lab6-1.asm  
~/work/study/2024-2025/Арх... пьютера/arch-pc/labs/lab06  
%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.3 Редактирование файла

Создаю исполняемый файл программы и запускаю его. Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6 (рис. 4).

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-1.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-1
lab6-1.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-1
j
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.4 Запуск исполняемого файла

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 5).



```
Открыть ▾ + • lab6-1.asm ~/work/study/2024-2025/Арх... пьютера/arch-pc/labs/lab06
%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.5 Редактирование файла

Создаю новый исполняемый файл программы и запускаю его. Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран (рис. 6).

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-1.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-1
lab6-1.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-1

shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.6 Запуск исполняемого файла

Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. 7).

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ touch lab6-2.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2.asm report
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.7 Создание нового файла

Ввожу в файл текст другой программы для вывода значения регистра `eax` (рис. 8).



```

%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit

```

Рис.8 Редактирование файла

Создаю и запускаю исполняемый файл `lab6-2`. Теперь выводится число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4” (рис. 9).

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-2.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-2
106
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.9 Запуск исполняемого файла

Заменяю в тексте программы в файле `lab6-2.asm` символы “6” и “4” на числа 6 и 4 (рис. 10).



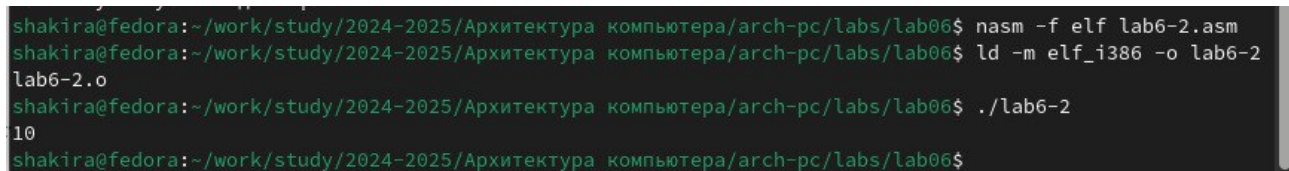


```
Открыть ▾ + • lab6-2.asm
~/work/study/2024-2025/Арх... пьютера/arch-pc/labs/lab06

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

Рис.10 Редактирование файла

Создаю и запускаю новый исполняемый файл. Теперь программа складывает не соответствующие символы коды в системе ASCII, а сами числа, поэтому вывод 10 (рис. 11).



```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-2.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-2
10
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.11 Запуск исполняемого файла

Заменяю в тексте программы функцию iprintLF на iprint (рис. 12).



```
Открыть ▾ + • lab6-2.asm
~/work/study/2024-2025/Арх... пьютера/arch-pc/labs/lab06

%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис.12 Редактирование файла

Создаю и запускаю новый исполняемый файл. Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией iprintLF, а iprint не добавляет к выводу символ переноса строки, в отличие от iprintLF (рис. 13).

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-2.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-2
lab6-2.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-2
10shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.13 Запуск исполняемого файла

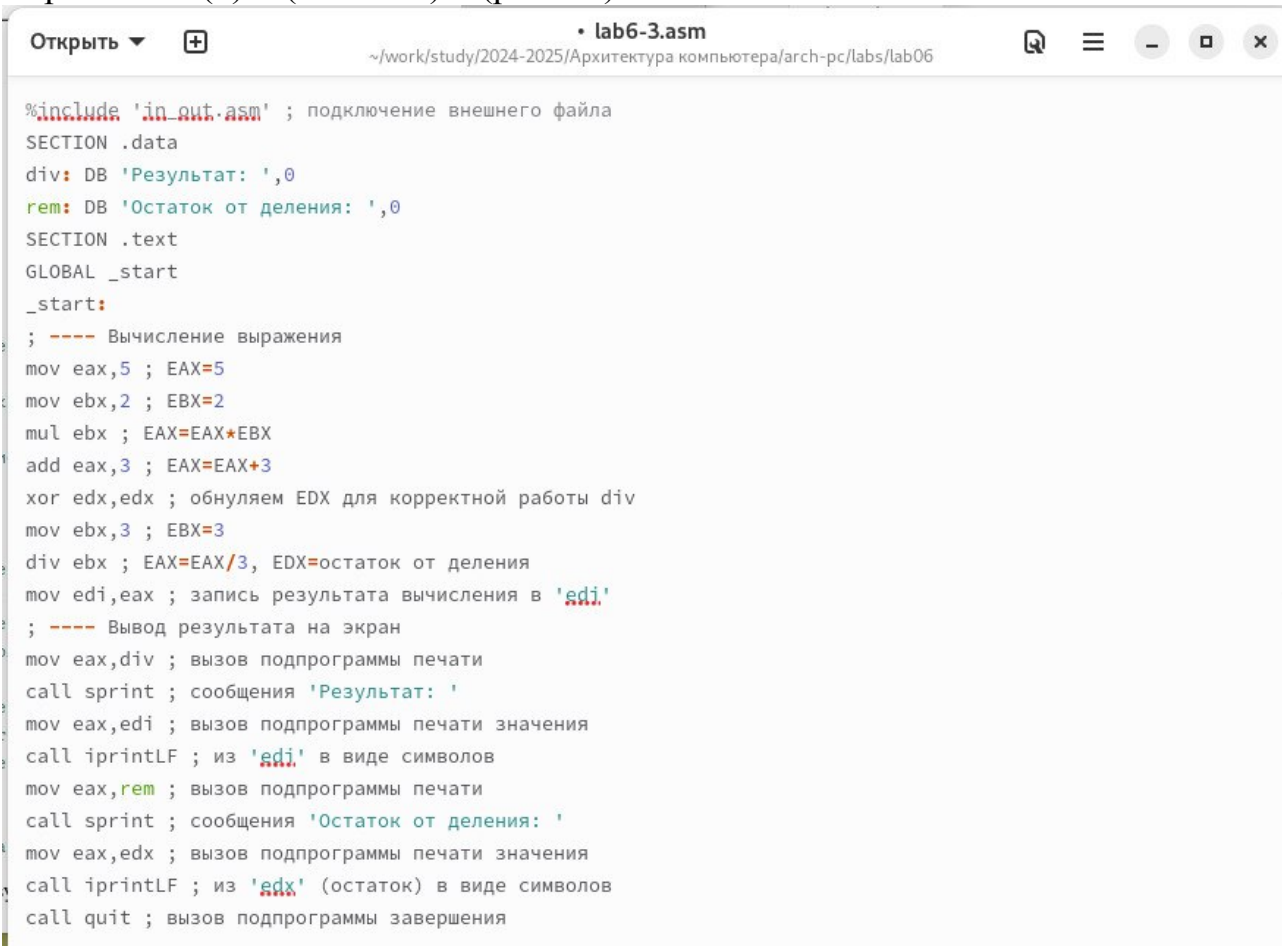
## 4.2 Выполнение арифметических операций в NASM

Создаю файл lab7-3.asm с помощью утилиты touch (рис. 14).

```
10shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ touch lab6-3.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2 lab6-2.asm lab6-2.o lab6-3.asm report
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.14 Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения  $f(x) = (5 * 2 + 3)/3$  (рис. 15).



```
Открыть ▾ + lab6-3.asm
~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06

%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис.15 Редактирование файла

Создаю исполняемый файл и запускаю его (рис. 16).

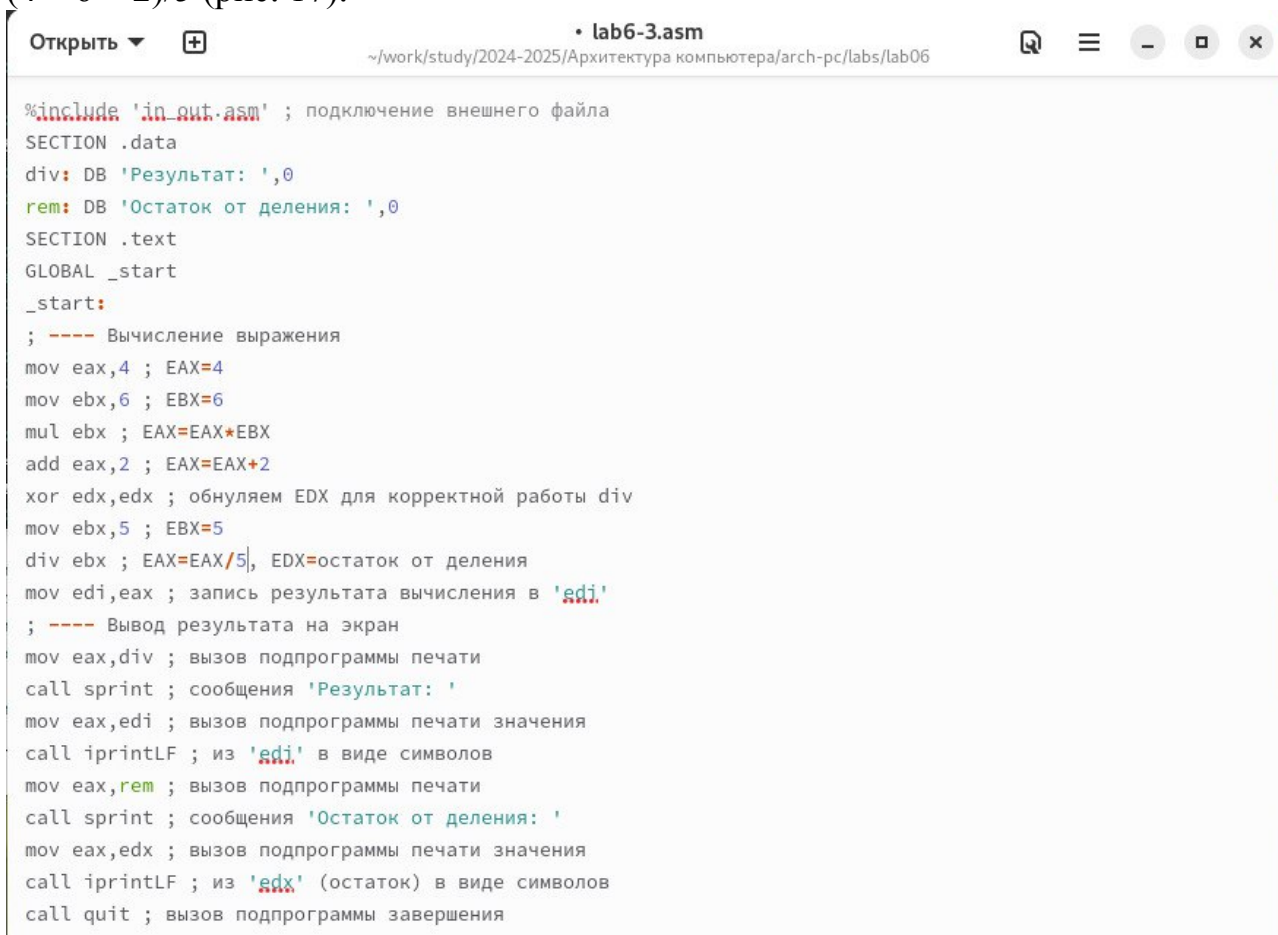
```

shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-3.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$

```

Рис.16 Запуск исполняемого файла

Изменяю программу так, чтобы она вычисляла значение выражения  $f(x) = (4 * 6 + 2)/5$  (рис. 17).



```

• lab6-3.asm
~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06

%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

Рис.17 Изменение программы

Создаю и запускаю новый исполняемый файл. Я посчитала для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно. (рис. 18).

```

shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-3.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$

```

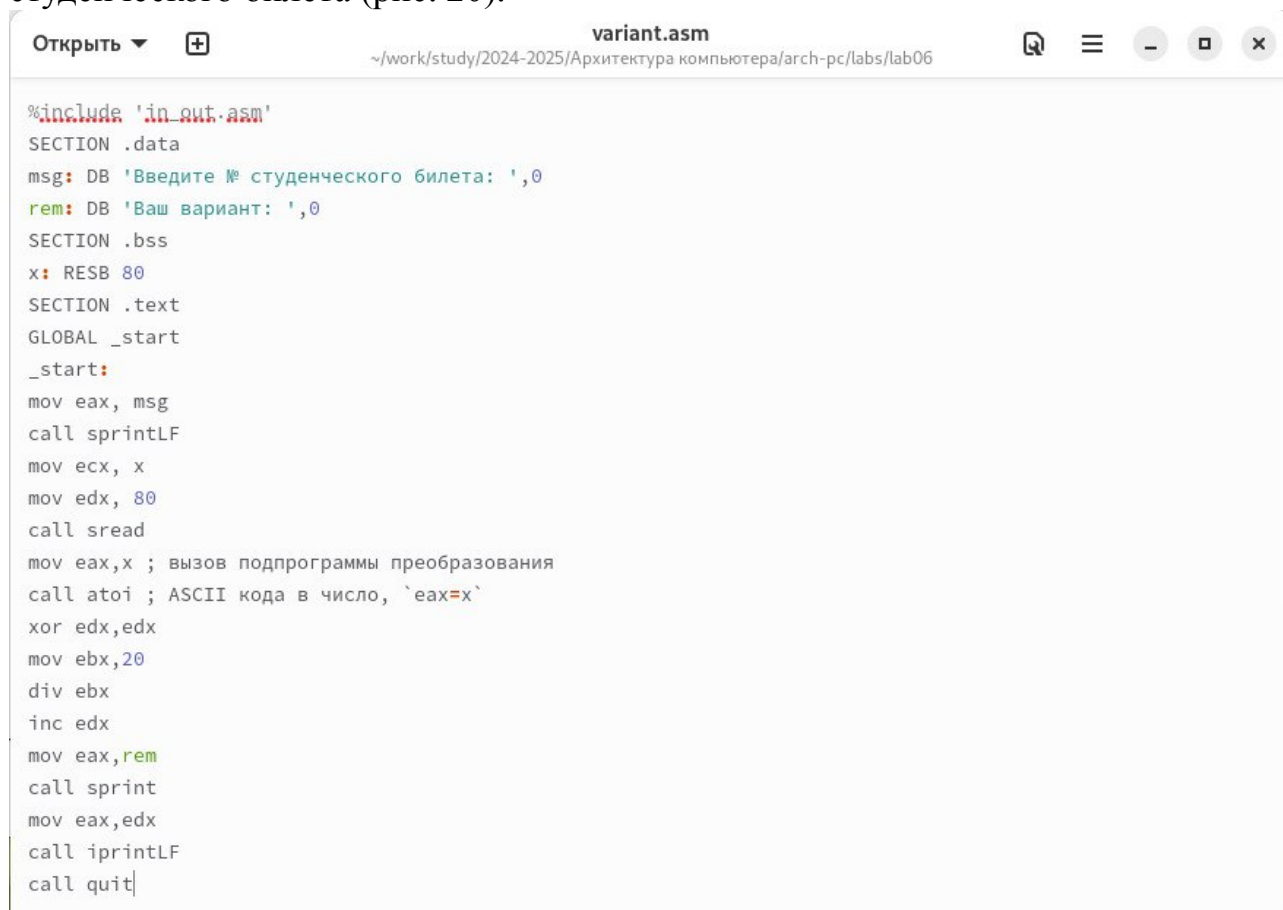
Рис.18 Исполнение файла

Создаю файл variant.asm с помощью утилиты touch (рис. 19).

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ touch variant.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ls
in_out.asm lab6-1.asm lab6-2 lab6-2.o lab6-3.asm report
lab6-1 lab6-1.o lab6-2.asm lab6-3 lab6-3.o variant.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.19 Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 20).



```
variant.asm
~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06

%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.20 Редактирование файла

Создаю и запускаю исполняемый файл (рис. 21). Ввожу номер своего студенческого билета, программа вывела, что мой вариант - 4

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf variant.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o variant
variant.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./variant
Введите № студенческого билета:
1132246743
Ваш вариант: 4
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.21 Запуск исполняемого файла

#### 4.2.1 Ответы на вопросы по программе

1. За вывод сообщения “Ваш вариант” отвечают строки кода:

```
mov eax, rem
call sprintf
```



2. Инструкция `mov ecx, x` используется для помещения адреса вводимой строки `x` в регистр `ecx`. Инструкция `mov edx, 80` задаёт длину вводимой строки, записывая значение 80 в регистр `edx`. Вызов подпрограммы `call sread` из внешнего файла обеспечивает ввод сообщения с клавиатуры. Инструкция `mov ecx, x` используется для помещения адреса вводимой строки `x` в регистр `ecx`. Инструкция `mov edx, 80` задаёт длину вводимой строки, записывая значение 80 в регистр `edx`. Вызов подпрограммы `call sread` из внешнего файла обеспечивает ввод сообщения с клавиатуры.
3. Инструкция `call atoi` вызывает подпрограмму из внешнего файла, которая преобразует символ в формате ASCII в целое число и сохраняет результат в регистре `eax`.
4. За вычисления варианта отвечают строки:

```
xor edx,edx ; обнуление edx для корректной работы div
```

```
mov ebx,20 ; ebx = 20
```

```
div ebx ; eax = eax/20, edx - остаток от деления
```

```
inc edx ; edx = edx + 1
```

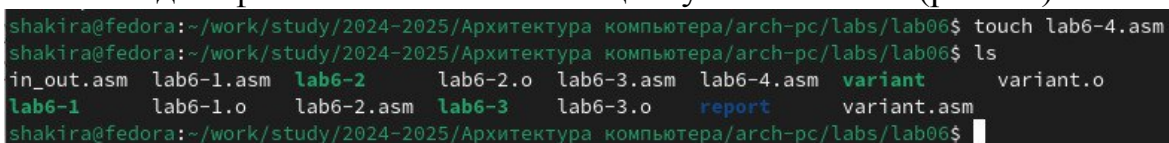
5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1
7. За вывод на экран результатов вычислений отвечают строки:

```
mov eax,edx
```

```
call iprintLF
```

### 4.3 Выполнение заданий для самостоятельной работы

Создаю файл `lab7-4.asm` с помощью утилиты `touch` (рис. 22).



```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ touch lab6-4.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ls
in_out.asm lab6-1.asm lab6-2 lab6-2.o lab6-3.asm lab6-4.asm variant variant.o
lab6-1 lab6-1.o lab6-2.asm lab6-3 lab6-3.o report variant.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.22 Создание файла

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения  $\frac{4}{3} \cdot (x - 1) + 5$ . Это выражение было под вариантом 4 (рис. 23).

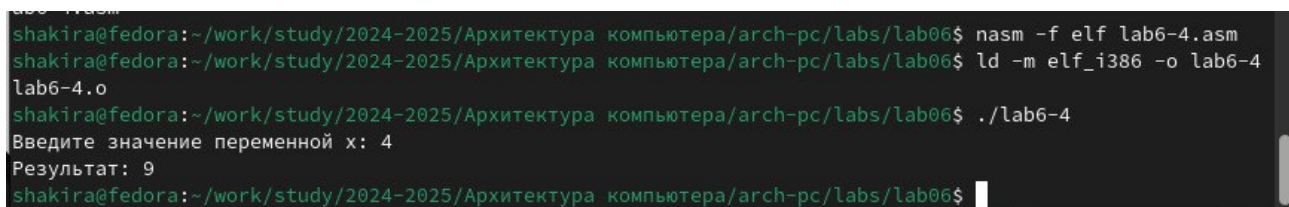


```
lab6-4.asm
~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06

%include 'in_out.asm'
SECTION .data
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
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; eax = eax - 1 = x - 1
mov ebx, 4
mul ebx ; eax = eax * 4 = 4 * (x - 1)
mov ebx, 3
div ebx ; eax = eax / 3 = 4/3 * (x - 1)
add eax, 5; eax = eax + 5 = 4/3 * (x-1) + 5
mov edi, eax ; запись результата вычисления в 'edi'
mov eax, rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
call quit
```

Рис.23 Написание программы

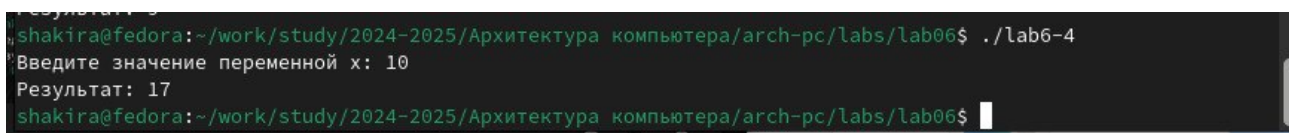
Создаю и запускаю исполняемый файл. При вводе значения 4, вывод - 9 (рис. 24).



```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ nasm -f elf lab6-4.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-4
Введите значение переменной x: 4
Результат: 9
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.24 Запуск исполняемого файла

Провожу еще один запуск исполняемого файла для проверки работы программы с другим значением на входе. Программа отработала верно (рис. 25).



```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$ ./lab6-4
Введите значение переменной x: 10
Результат: 17
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab06$
```

Рис.25 Запуск исполняемого файла

Код программы:

```
%include 'in_out.asm'

SECTION .data
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start

_start:
mov eax, msg
call sprint
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;  $eax = eax - 1 = x - 1$ 
mov ebx,4
mul ebx ;  $eax = eax * 4 = 4 * (x - 1)$ 
mov ebx,3
div ebx ;  $eax = eax / 3 = 4/3 * (x - 1)$ 
add eax,5;  $eax = eax + 5 = 4/3 * (x-1) + 5$ 
mov edi,eax ; запись результата вычисления в 'edi'
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
call quit
```

## **5 Выводы**

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.



## **6 Источники**

1. [Архитектура ЭВМ \(rudn.ru\)](http://rudn.ru)
2. Таблица ASCII