

# Отчет по лабораторной работе №6

## Дисциплина: архитектура компьютера

Акмурадов Тимур

### Содержание

Цель работы .....	1
Задание .....	1
Теоретическое введение .....	1
Выполнение лабораторной работы .....	2
Символьные и численные данные в NASM .....	2
Выполнение арифметических операций в NASM .....	6
Ответы на вопросы по программе .....	9
Выполнение заданий для самостоятельной работы .....	10
Выводы.....	12
Список литературы .....	12

### Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM.

### Задание

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

### 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. - Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: `mov ax,bx`. - Непосредственная адресация – значение операнда задается непосредственно в команде, Например: `mov ax,2`. - Адресация памяти – операнд задает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

Ввод информации с клавиатуры и вывод её на экран осуществляется в символьном виде. Кодирование этой информации производится согласно кодовой таблице символов ASCII. ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно стандарту ASCII каждый символ кодируется одним байтом. Среди инструкций NASM нет такой, которая выводит числа (не в символьном виде). Поэтому, например, чтобы вывести число, надо предварительно преобразовать его цифры в ASCII-коды этих цифр и выводить на экран эти коды, а не само число. Если же выводить число на экран непосредственно, то экран воспримет его не как число, а как последовательность ASCII-символов – каждый байт числа будет воспринят как один ASCII-символ – и выведет на экран эти символы. Аналогичная ситуация происходит и при вводе данных с клавиатуры. Введенные данные будут представлять собой символы, что сделает невозможным получение корректного результата при выполнении над ними арифметических операций. Для решения этой проблемы необходимо проводить преобразование ASCII символов в числа и обратно

## Выполнение лабораторной работы

### Символьные и численные данные в NASM

С помощью утилиты `mkdir` создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6 (рис. 01). Перехожу в созданный каталог с помощью утилиты `cd`.

```
takmuradow@Linux:~$ mkdir ~/work/study/2023-2024/"Computer architecture"/arch-pc/lab06
takmuradow@Linux:~$ cd /work/study/2023-2024/"Computer architecture"/arch-pc/lab06
bash: cd: /work/study/2023-2024/Computer architecture/arch-pc/lab06: No such file or directory
takmuradow@Linux:~$ cd work/study/2023-2024/"Computer architecture"/arch-pc/lab06
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
```

С помощью утилиты `touch` создаю файл `lab6-1.asm` (рис. 02).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$ touch lab6-1.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$ ls
lab6-1.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
```

Копирую в текущий каталог файл `in_out.asm` с помощью утилиты `cp`, т.к. он будет использоваться в других программах (рис.03).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$ cp ~/Downloads/in_out.asm in_out.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$ ls
in_out.asm lab6-1.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
```

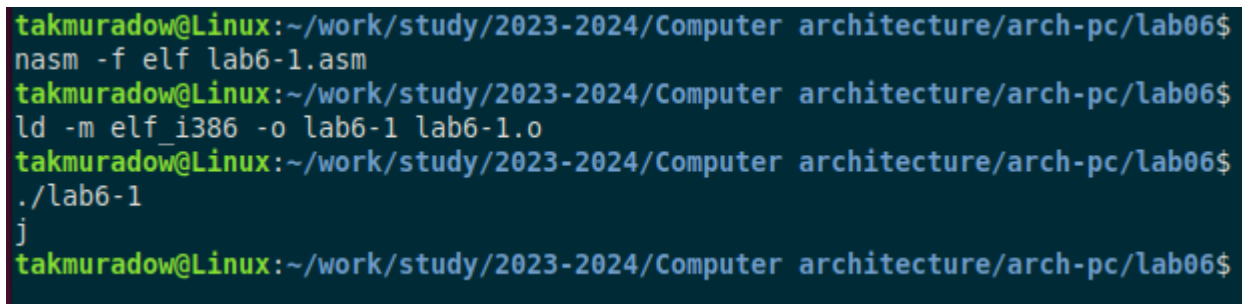
Открываю созданный файл `lab6-1.asm`, вставляю в него программу вывода значения регистра `eax` (рис. 04).



```
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax, '6'
8 mov ebx, '4'
9 add eax, ebx
10 mov [buf1],eax
11 mov eax,buf1
12 cal sprintf
13 call quit
```

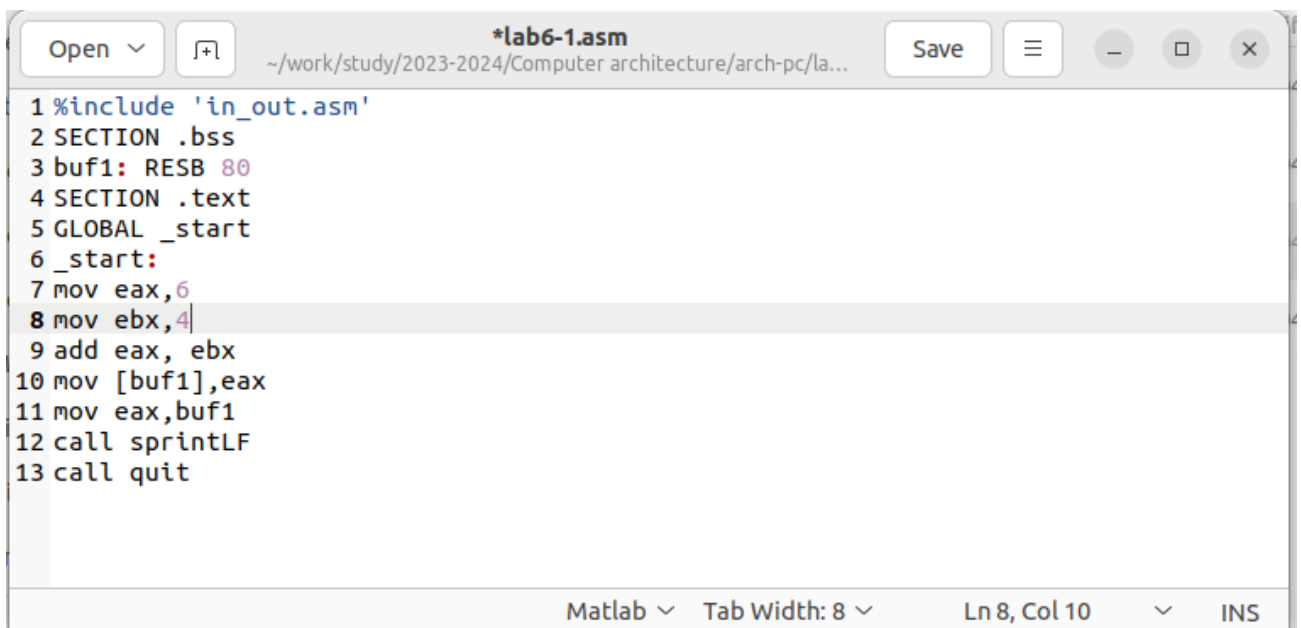
Plain Text ▾ Tab Width: 8 ▾ Ln 13, Col 10 ▾ INS

Создаю исполняемый файл программы и запускаю его (рис. 05). Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.



```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
nasm -f elf lab6-1.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
ld -m elf_i386 -o lab6-1 lab6-1.o
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
./lab6-1
j
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
```

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 06).



```
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,6
8 mov ebx,4
9 add eax, ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintf
13 call quit
```

Matlab ▾ Tab Width: 8 ▾ Ln 8, Col 10 ▾ INS

Создаю новый исполняемый файл программы и запускаю его (рис. 07). Теперь вывелся


символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$  
nasm -f elf lab6-1.asm  
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$  
ld -m elf_i386 -o lab6-1 lab6-1.o  
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$  
./lab6-1  
  
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
```

Создаю новый файл lab6-2.asm с помощью утилиты touch (рис. 08).

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$  
touch lab6-2.asm  
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$  
ls  
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2.asm  
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06$
```

Ввожу в файл текст другой программы для вывода значения регистра eax (рис. 09).



```
Open  [icon] *lab6-2.asm Save [icon] [icon] [icon]  
~/work/study/2023-2024/Computer architecture/arch-pc/la...  
1 %include 'in_out.asm'  
2 SECTION .text  
3 GLOBAL _start  
4 _start:  
5 mov eax,'6'  
6 mov ebx,'4'  
7 add eax,ebx  
8 call iprintLF  
9 call quit  
  
Plain Text  Tab Width: 8  Ln 9, Col 10  INS
```

Создаю и запускаю исполняемый файл lab6-2 (рис. 10). Теперь вывод число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”.

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ nasm -f elf lab6-2.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ld -m elf_i386 -o lab6-2 lab6-2.o
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ./lab6-2
106
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$
```

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 11).



```
lab6-2.asm
~/work/study/2023-2024/Computer architecture/arch-pc/la...
Open [icon] Save [icon] [icon] [icon] [icon]
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit
Loading file "/home/takmuradow/work/stu... Matlab Tab Width: 8 Ln 9, Col 10 INS
```

Создаю и запускаю новый исполняемый файл (рис. 12). Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ nasm -f elf lab6-2.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ld -m elf_i386 -o lab6-2 lab6-2.o
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ./lab6-2
10
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$
```

Заменяю в тексте программы функцию iprintLF на iprint (рис. 13).

```
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprint
9 call quit
```

Создаю и запускаю новый исполняемый файл (рис. 14). 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`.

```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ nasm -f elf lab6-2.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ld -m elf_i386 -o lab6-2 lab6-2.o
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ./lab6-2
$
10
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$
```

### Выполнение арифметических операций в NASM

Создаю файл `lab6-3.asm` с помощью утилиты `touch` (рис. 15).

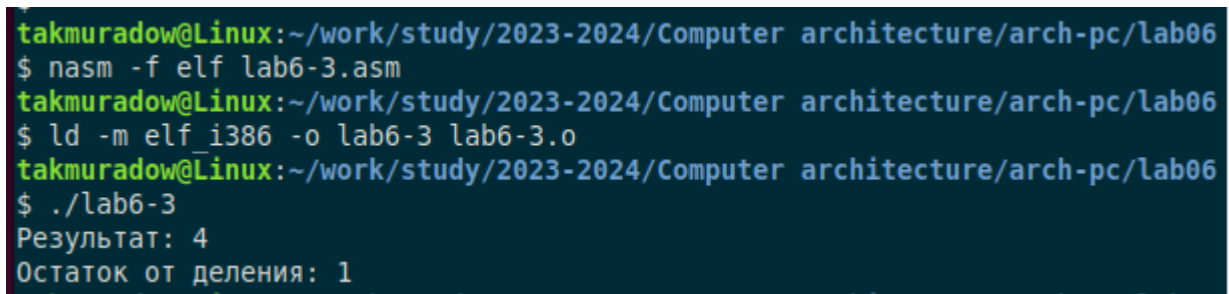
```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ touch lab6-3.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ls
in_out.asm  lab6-1.asm  lab6-2      lab6-2.o
lab6-1      lab6-1.o    lab6-2.asm  lab6-3.asm
```

Ввожу в созданный файл текст программы для вычисления значения выражения  $f(x) = (5 * 2 + 3)/3$  (рис. 16).



```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax,5
9 mov ebx,2
10 mul ebx
11 add eax,3
12 xor edx,edx
13 mov ebx,3
14 div ebx
15 mov edi,eax
16 mov eax,edi
17 call sprint
18 mov eax,edi
19 call iprintLF
20 mov eax,rem
21 call sprint
22 mov eax,edx
23 call iprintLF
24 call quit
```

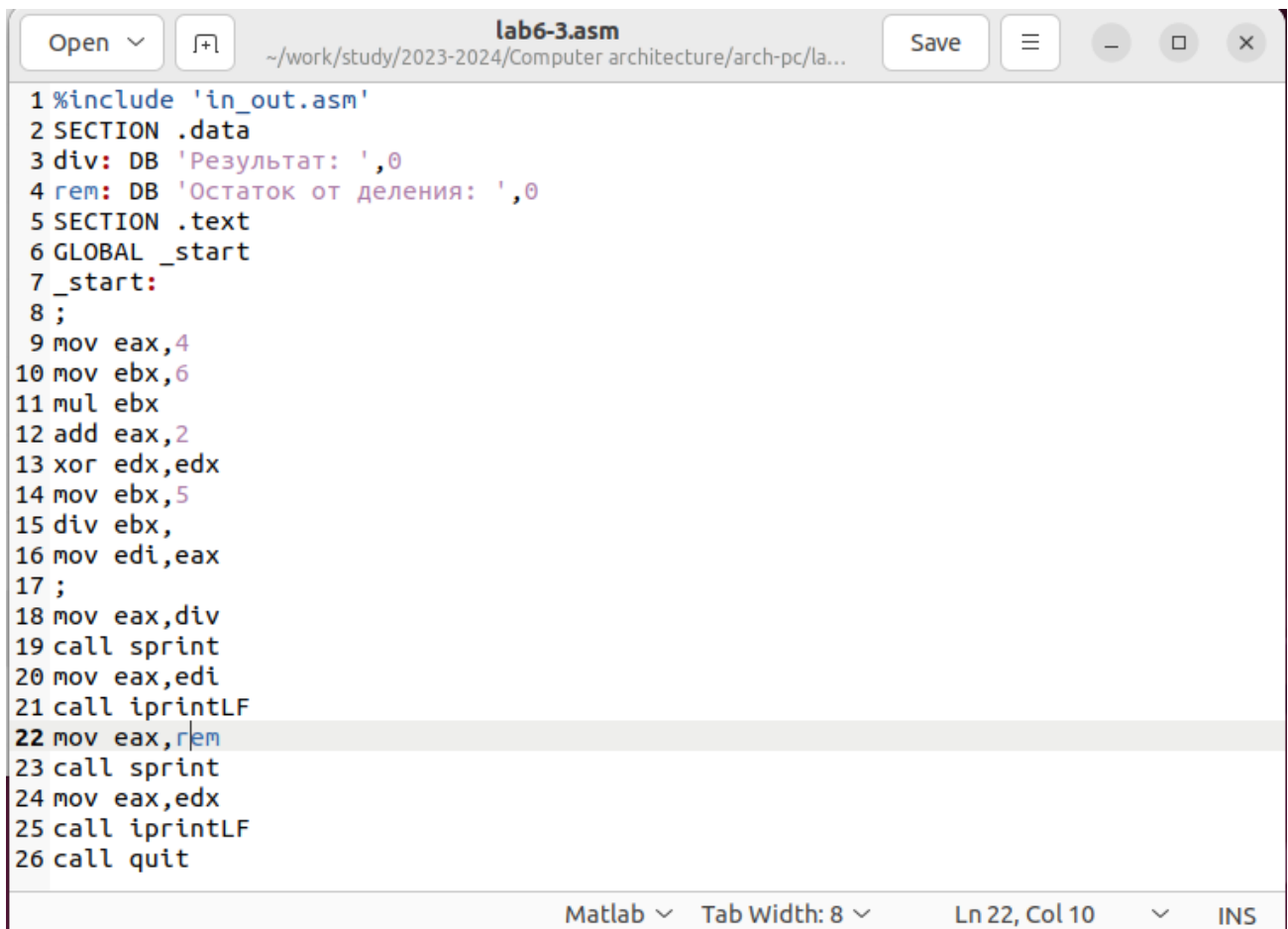
Создаю исполняемый файл и запускаю его (рис. 17).



```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ nasm -f elf lab6-3.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ld -m elf_i386 -o lab6-3 lab6-3.o
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Изменяю программу так, чтобы она вычисляла значение выражения  $f(x) = (4 * 6 + 2)/5$  (рис. 18).

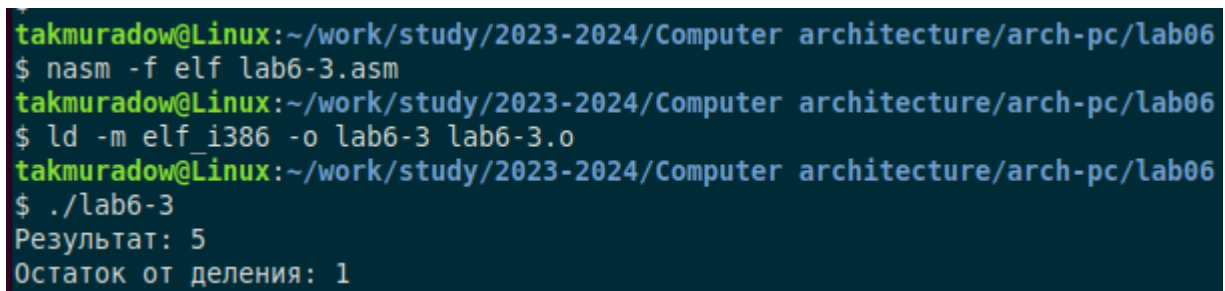




```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ;
9 mov eax,4
10 mov ebx,6
11 mul ebx
12 add eax,2
13 xor edx,edx
14 mov ebx,5
15 div ebx
16 mov edi,eax
17 ;
18 mov eax,div
19 call sprint
20 mov eax,edi
21 call iprintLF
22 mov eax,rem
23 call sprint
24 mov eax,edx
25 call iprintLF
26 call quit
```

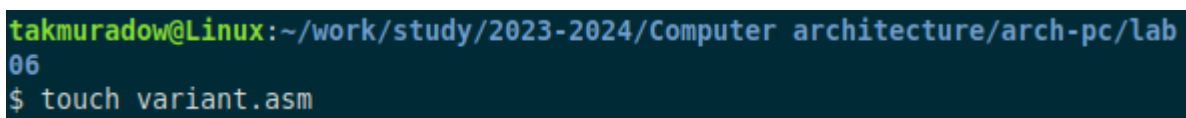
Matlab ▾ Tab Width: 8 ▾ Ln 22, Col 10 ▾ INS

Создаю и запускаю новый исполняемый файл (рис. 19). Я посчитал для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно.



```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ nasm -f elf lab6-3.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ld -m elf_i386 -o lab6-3 lab6-3.o
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Создаю файл variant.asm с помощью утилиты touch (рис. 20).



```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ touch variant.asm
```

Ввожу

в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 21).



```

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите номер студенческого билета: ',0
4 rem: DB 'Ваш вариант: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax,msg
11 call sprintfLF
12 mov ecx,x
13 mov edx, 80
14 call sread
15 mov eax,x
16 call atoi
17 xor edx,edx
18 mov ebx,20
19 div ebx
20 inc edx
21 mov eax,rem
22 call sprintf
23 mov eax,edx
24 call iprintLF
25 call quit

```

Saving file "/home/takmuradow/work/stud... Matlab ▾ Tab Width: 8 ▾ Ln 25, Col 10 ▾ INS

Создаю и запускаю исполняемый файл (рис. 22). Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 8.

```

takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ nasm -f elf variant.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ld -m elf_i386 -o variant variant.o
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ./variant
Введите номер студенческого билета:
1032234240
Ваш вариант: 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, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры

3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`
4. За вычисления варианта отвечают строки:

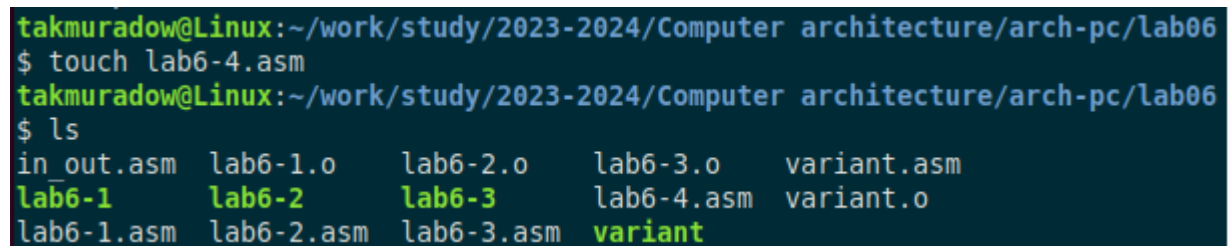
```
xor edx,edx ; обнуление edx для корректной работы div
mov ebx,20 ; ebx = 20
div ebx ; eax = eax/20, edx - остаток от деления
inc edx ; edx = edx + 1
```

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
```

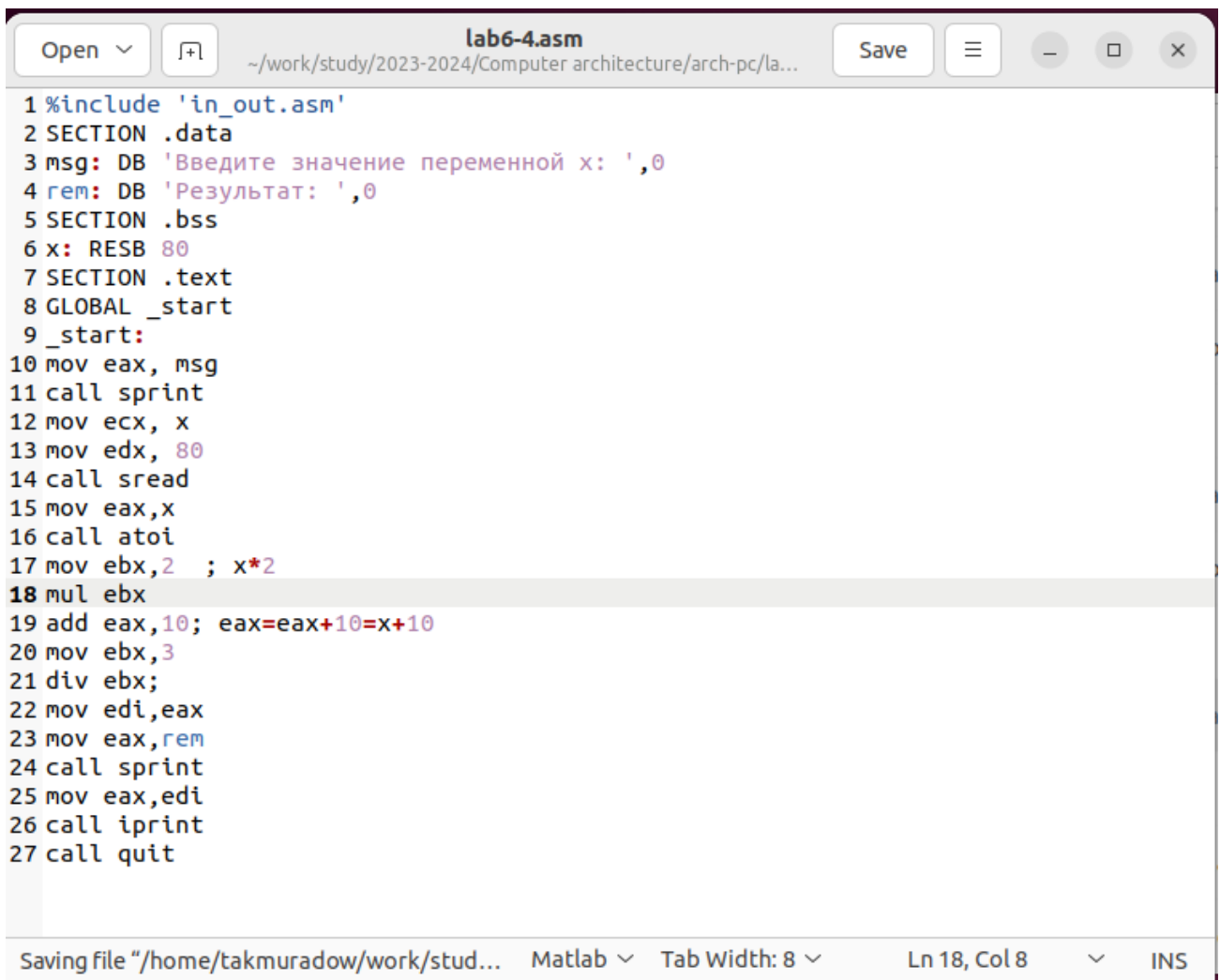
## Выполнение заданий для самостоятельной работы

Создаю файл `lab7-4.asm` с помощью утилиты `touch` (рис. 23).



```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ touch lab6-4.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ls
in_out.asm  lab6-1.o    lab6-2.o    lab6-3.o    variant.asm
lab6-1      lab6-2      lab6-3      lab6-4.asm  variant.o
lab6-1.asm  lab6-2.asm  lab6-3.asm  variant
```

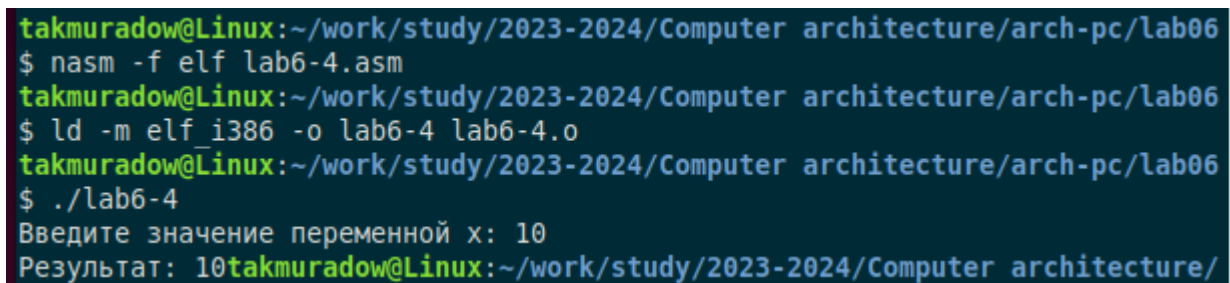
Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения  $(10 + 2x) / 3$  (рис. 24). Это выражение было под вариантом 1.



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите значение переменной x: ',0
4 rem: DB 'Результат: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprint
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax, x
16 call atoi
17 mov ebx, 2 ; x*2
18 mul ebx
19 add eax, 10; eax=eax+10=x+10
20 mov ebx, 3
21 div ebx;
22 mov edi, eax
23 mov eax, rem
24 call sprint
25 mov eax, edi
26 call iprint
27 call quit
```

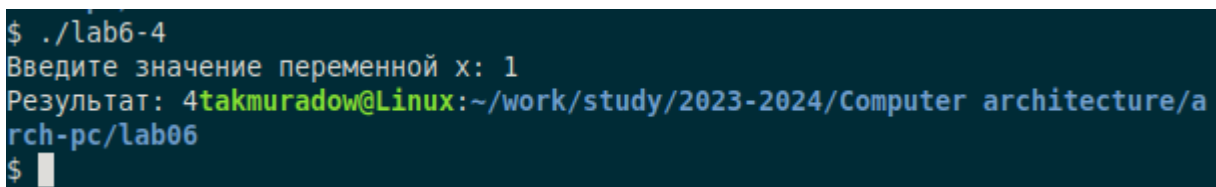
Saving file "/home/takmuradow/work/stud... Matlab Tab Width: 8 Ln 18, Col 8 INS

Создаю и запускаю исполняемый файл (рис. 25). При вводе значения 10, вывод - 10.



```
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ nasm -f elf lab6-4.asm
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ld -m elf_i386 -o lab6-4 lab6-4.o
takmuradow@Linux:~/work/study/2023-2024/Computer architecture/arch-pc/lab06
$ ./lab6-4
Введите значение переменной x: 10
Результат: 10takmuradow@Linux:~/work/study/2023-2024/Computer architecture/
```

Провожу еще один запуск исполняемого файла для проверки работы программы с другим значением на входе (рис. 26). Программа отработала верно.



```
$ ./lab6-4
Введите значение переменной x: 1
Результат: 4takmuradow@Linux:~/work/study/2023-2024/Computer architecture/a
rch-pc/lab06
$
```

#### Листинг 4.1. Программа для вычисления значения выражения $(10 + 2x) / 3$ .

```
%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
call atoi
mov ebx, 2 ; x*2
mul ebx
add eax, 10; eax=eax+10=x+10
mov ebx, 3
div ebx;
mov edi, eax
mov eax, rem
call sprint
mov eax, edi
call iprint
call quit
```

## Выводы

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

## Список литературы

1. Лабораторная работа №6
2. Таблица ASCII