

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**  
**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЁТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**  
**Дисциплина: Архитектура компьютера**

Студент: Ванюшкина Т.В.

Группа: НКАбд-01-24

Студ.билет: 1132246713

**МОСКВА**

**2024г**

# **СОДЕРЖАНИЕ**

<b>1. Цель работы.....</b>	<b>3</b>
<b>2. Теоретическое введение.....</b>	<b>4</b>
<b>3. Выполнение лабораторной работы.....</b>	<b>5-14</b>
<b>3.1 Символьные и численные данные в NASM.....</b>	<b>5-9</b>
<b>3.2 Выполнение арифметических операций в NASM.....</b>	<b>9-12</b>
<b>3.3 Задание для самостоятельной работы.....</b>	<b>13-14</b>
<b>4. Выводы.....</b>	<b>15</b>
<b>5. Список литературы.....</b>	<b>16</b>

## **1. Цель работы**

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

## **2. Теоретическое введение**

Язык ассемблера NASM (Netwide Assembler) - это мощный инструмент для работы с аппаратным обеспечением и создания низкоуровневого кода. В основе его возможностей лежат арифметические инструкции, которые позволяют выполнять основные математические операции над данными. В этом докладе мы рассмотрим ключевые арифметические инструкции NASM, их синтаксис, примеры использования и нюансы.

## 3.Выполнение лабораторной работы

### 3.1 Символьные и численные данные в NASM

Создаю каталог для программ лабораторной работы № 6, перехожу в него и создаю файл lab6-1.asm:

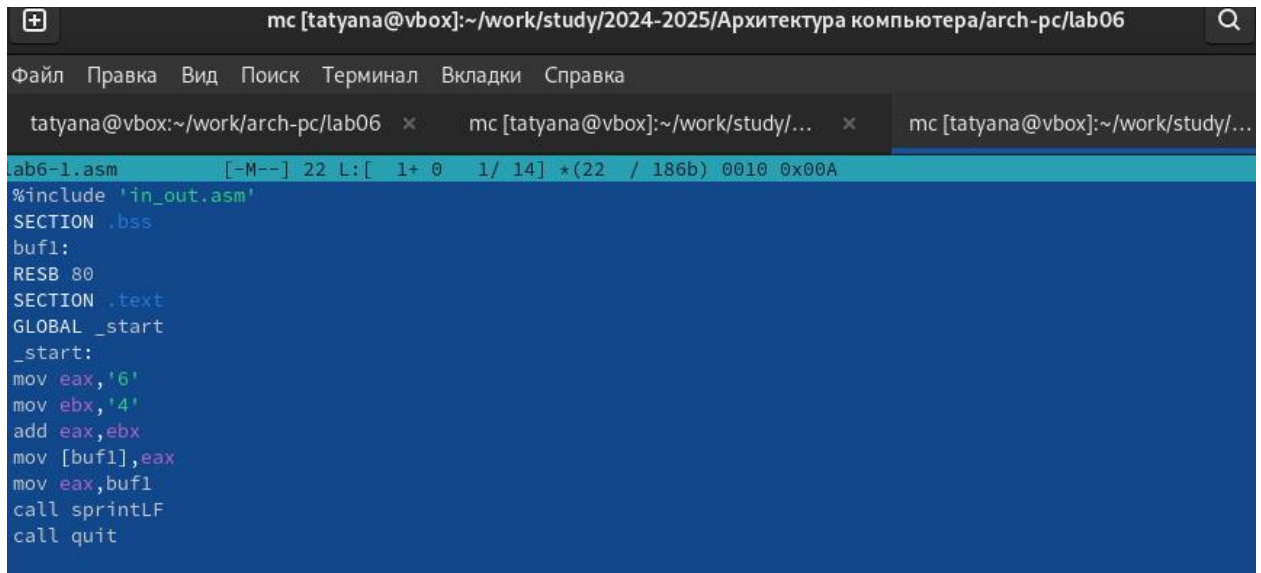
```
tatyana@vbox:~/work/arch-pc/lab06$ mkdir ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab06
```

Рис.1: Создание каталога

```
tatyana@vbox:~/work/arch-pc/lab06$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab06
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-1.asm
```

Рис.2: Создание файла

Ввожу в файл lab6-1.asm текст программы из листинга 6.1.:



```
mc [tatyana@vbox]:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06
Файл Правка Вид Поиск Терминал Вкладки Справка
tatyana@vbox:~/work/arch-pc/lab06 x mc [tatyana@vbox]:~/work/study/... x mc [tatyana@vbox]:~/work/study/...
lab6-1.asm [-M--] 22 L:[ 1+ 0 1/ 14] *(22 / 186b) 0010 0x00A
#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 3:Ввод текста программы

Создаю исполняемый файл и запускаю его:

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

Рис.4:Создание и запуск файла

Исправляю текст программы (Листинг6.1) следующим образом:  
заменяю строки:

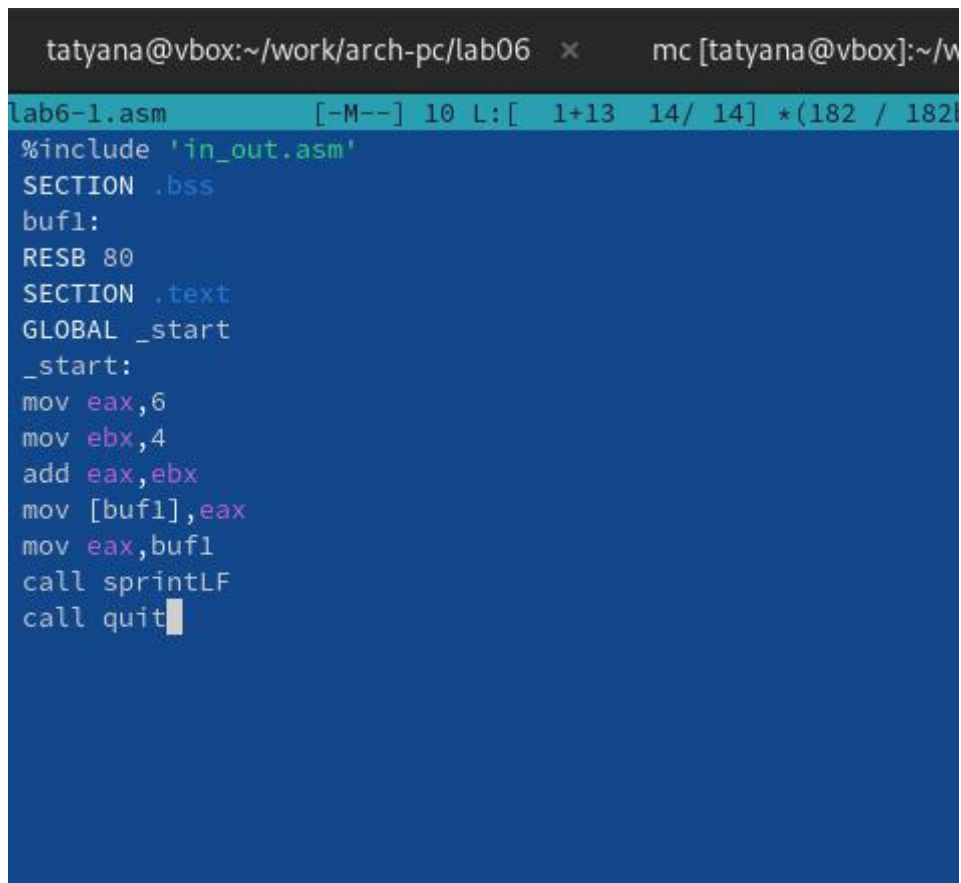
```
mov eax,'6'
```

```
mov ebx,'4'
```

на строки :

```
mov eax,6
```

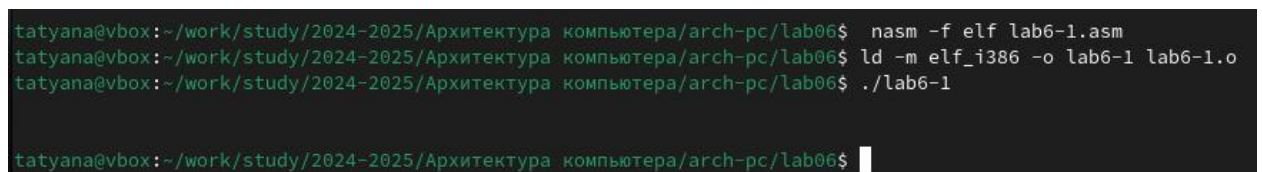
```
mov ebx,4
```



```
tatyana@vbox:~/work/arch-pc/lab06 x mc [tatyana@vbox]:~/w
lab6-1.asm [-M--] 10 L:[ 1+13 14/ 14] *(182 / 182)
#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: Замена текста программы

Создаю исполняемый файл и запускаю его:



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

tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

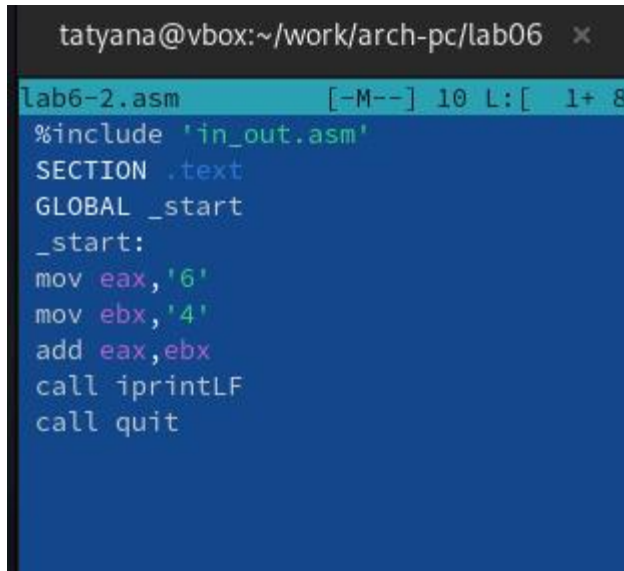
Рис.6 : Создание и запуск файла

Создаю файл lab6-2.asm:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-2.asm
```

Рис.7:Создание файла

Ввожу в него текст программы из листинга 6.2.:



```
lab6-2.asm [-M--] 10 L: [ 1+ 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: Ввод текста программы

Создаю исполняемый файл и запускаю его:

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

Рис.9: Создание и запуск файла

Аналогично предыдущему примеру изменяю символы на числа.  
Заменяю строки:

```
mov eax, '6'
```

```
mov ebx, '4'
```

на строки:

```
mov eax, 6
```

```
mov ebx, 4
```

```
tatyana@vbox:~/work/... x mc [ta
lab6-2.asm [-M--] 10 L: [ 1+
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис.10: Замена текста программы

Создаю исполняемый файл и запускаю его:

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

Рис.11: Создание и запуск файла

Заменяю функцию iprintLF на iprint:

```
tatyana@vbox:~/work/... x mc [ta
lab6-2.asm [-M--] 10 L: [ 1+
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис.12: Замена текста программы



Создаю исполняемый файл и запускаю его:

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

Рис.13: Создание и запуск файла

## 3.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm:

```
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-3.asm
```

Рис.14: Создание файла

Ввожу в новый файл текст программы из листинга 6.3:

```
tatyana@vbox:~/work/... x mc [tatyana@vbox]:~/... x tatyana@vbox:~/work/... x
lab6-3.asm [-----] 1 L: [ 1+24 25/ 25] *(375 / 375b) <E
%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
mov eax,edx
call iprintLF
call quit
```

Рис.15: Ввод текста программы

Создаю исполняемый файл и запускаю его:

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

Рис.16: Создание и запуск файла

Изменяю текст программы для вычисления  $f(x) = (4 * 6 + 2)/5$ :

```
tatyana@vbox:~/work/... x mc [tatyana@vbox]:~/... x tatyana@vbox:~/v
lab6-3.asm [-M--] 10 L:[ 1+12 13/ 25] *(232 / 375b) 0010 0x00A
%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
mov eax,edx
call iprintLF
call quit
```

Рис.17: Замена текста программы

Создаю исполняемый файл и проверяю его работу:

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

Рис.18: Создание и проверка файла

Создаю файл variant.asm:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch variant.asm
```

Рис.19: Создание файла

Ввожу в новый файл текст из листинга 6.4 :

```
variant.asm [-M--] 10 L:[ 1+24 25/ 25] *(523 / 523b) <EOF>
%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 sprintLF
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 sprint
mov eax, edx
call iprintLF
call quit
```

Рис.20: Ввод текста файла

Создаю исполняемый файл и запускаю его:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf variant.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246713
Ваш вариант: 14
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

Рис.21: Создание и запуск файла

Ответы на вопросы:

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения 'Ваш вариант:'?

ответ: за вывод на экран сообщения "Ваш вариант" отвечают строки

```
mov eax,rem
```

```
call sprint
```

2. Для чего используются следующие инструкции?

```
mov ecx, x
```

```
mov edx, 80
```

```
call sread
```

ответ: инструкция `mov ecx, x` используется для того, чтобы положить адрес вводимой строки `x` в регистр

инструкция `mov edx, 80` - это инструкция для записи в регистр `edx` длины вводимой строки

инструкция `call sread` - вызывает подпрограмму из внешнего файла, обеспечивающую ввод сообщения с клавиатуры

3. Для чего используется инструкция "call atoi"?

ответ: инструкция `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

ответ: `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. В какой регистр записывается остаток от деления при выполнении инструкции "div ebx"?

ответ: остаток от деления записывается в регистр `edx`

6. Для чего используется инструкция "inc edx"?

ответ: инструкция `inc edx` увеличивает значение регистра `edx` на 1

7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

ответ: за вывод на экран результата вычислений отвечают строки: `mov eax,edx` `call iprintLF`

### 3.3 Задание для самостоятельной работы

В каталоге создаю файл lab6-4.asm:

```
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch lab6-4.asm
```

Рис.22: Создание файла

Создаю программу для вычисления варианта:

```
lab6-4.asm      [-M--] 10 L: [ 1+24 25/ 25] *(408 / 408)
#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 sprintLF
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 sprint
mov eax, edx
call iprintLF
call quit
```

Рис.23: Создание программы

Создаю исполняемый файл и запускаю его:

```
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf lab6-4.asm
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./lab6-4
Введите № студенческого билета:
1132246713
Ваш вариант: 14
tatyana@vbox: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$
```

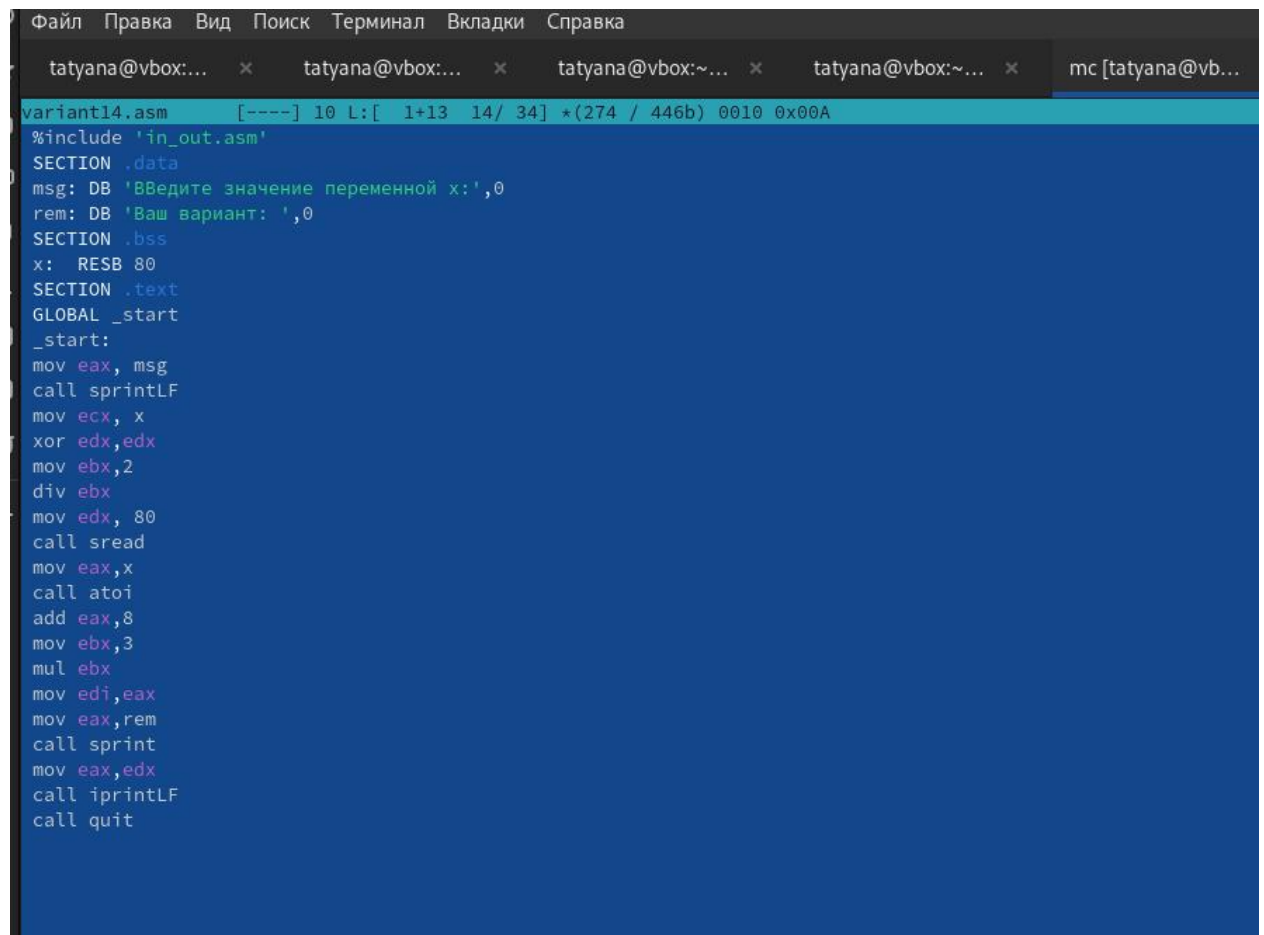
Рис.24: Создание и запуск файла

Создаю файл variant14.asm:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ touch variant14.asm
```

Рис.25: Создание файла

Ввожу в него текст программы для вычисления выражения 14)  $(x/2+8) \cdot 3$ :



```
variant14.asm [----] 10 L: [ 1+13 14/ 34] *(274 / 446b) 0010 0x00A
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите значение переменной x:',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
xor edx, edx
mov ebx, 2
div ebx
mov edx, 80
call sread
mov eax, x
call atoi
add eax, 8
mov ebx, 3
mul ebx
mov edi, eax
mov eax, rem
call sprintf
mov eax, edx
call iprintLF
call quit
```

Рис.26: Ввод текста программы

Создаю исполняемый файл и запускаю его:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ nasm -f elf variant14.asm
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ld -m elf_i386 -o variant14 variant14.o
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./variant14
Введите значение переменной x:
1
Результат: 0
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab06$ ./variant14
Введите значение переменной x:
4
Результат: 0
```

Рис.27: Создание и запуск файла

## **4. Выводы**

В ходе выполнения лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

## **5. Список литературы**

Курс: Архитектура компьютеров и операционные системы. Раздел  
"Архитектура компьютеров" (02.03.00, УГСН) (rudn.ru)