

Отчет по лабораторной работе №7

дисциплина: Архитектура компьютера

Галацан Николай, НПИбд-01-22

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выполнение заданий для самостоятельной работы	13
4	Выводы	16

Список иллюстраций

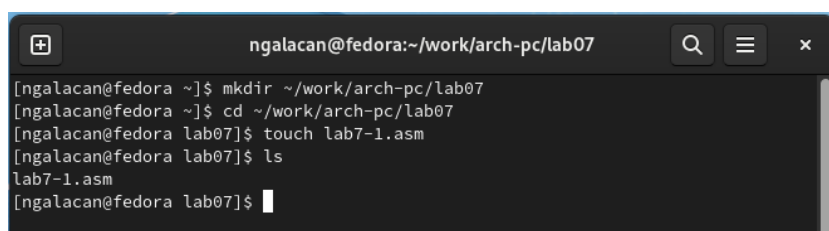
2.1	Создание каталога и файла lab7-1.asm	5
2.2	Текст программы lab7-1.asm	5
2.3	Трансляция, компоновка, запуск исполняемого файла lab 7-1 . . .	6
2.4	Трансляция, компоновка, запуск измененного исполняемого файла lab 7-1	6
2.5	Текст программы lab 7-2	7
2.6	Трансляция, компоновка, запуск исполняемого файла lab 7-2 . . .	7
2.7	Трансляция, компоновка, запуск измененного исполняемого файла lab 7-2	8
2.8	Результат замены функции iprintLF на iprint в lab7-2	8
2.9	Текст программы lab7-3.asm	9
2.10	Трансляция, компоновка, запуск исполняемого файла lab7-3 . . .	9
2.11	Измененный текст программы lab7-3.asm	10
2.12	Трансляция, компоновка, запуск измененного исполняемого файла lab7-3	10
2.13	Текст программы variant.asm	11
2.14	Трансляция, компоновка, запуск исполняемого файла variant . . .	11
3.1	Трансляция, компоновка, запуск исполняемого файла lab7-sam . .	15

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

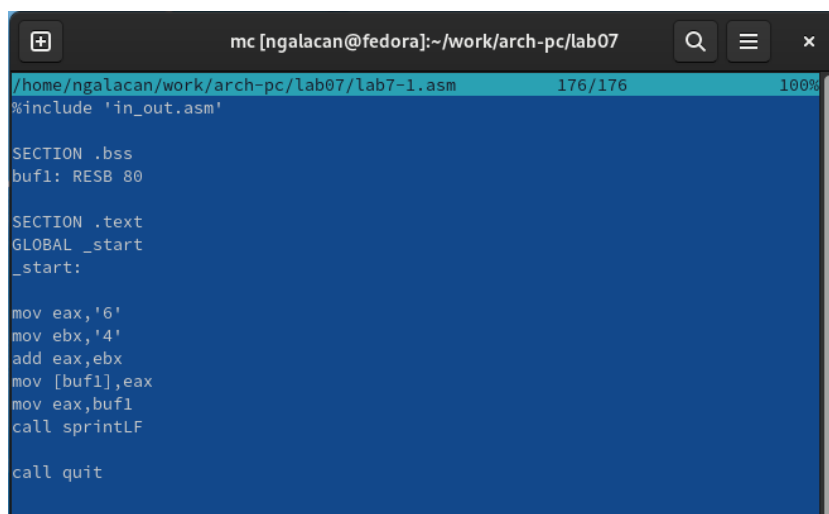
1. Ввожу команды для создания каталога лабораторной работы, перехожу в него, создаю файл `lab7-1.asm` и проверяю его наличие (рис. 2.1).



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

Рис. 2.1: Создание каталога и файла `lab7-1.asm`

2. С помощью функциональной клавиши **F4** открываю файл для редактирования, ввожу в файл `lab7-1.asm` текст программы из листинга 7.1, сохраняю файл и открываю для просмотра с помощью **F3** (рис. 2.2).



```
mc [ngalacan@fedora]:~/work/arch-pc/lab07
/home/ngalacan/work/arch-pc/lab07/lab7-1.asm 176/176 100%
#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. 2.2: Текст программы `lab7-1.asm`

Создаю исполняемый файл и запускаю его, предварительно скопировав файл `in_out.asm` в соответствующий каталог (рис. 2.3).

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

Рис. 2.3: Трансляция, компоновка, запуск исполняемого файла lab 7-1

В результате исполнения программы выводится символ *j*.

3. Далее изменяю текст программы и вместо символов записываю в регистры числа, заменив строки

```
mov eax, '6'
mov ebx, '4'
```

на

```
mov eax, 6
mov ebx, 4
```

Создаю исполняемый файл и запускаю его (рис. 2.4).

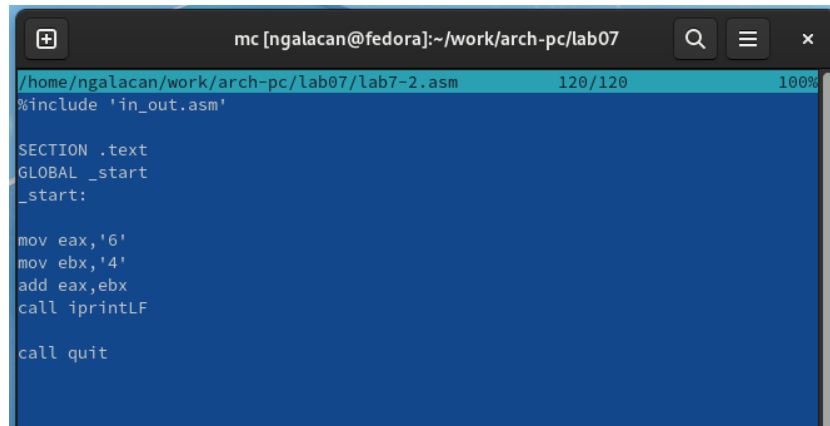
```
[ngalacan@fedora lab07]$ nasm -f elf lab7-1.asm
[ngalacan@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[ngalacan@fedora lab07]$ ./lab7-1

[ngalacan@fedora lab07]$
```

Рис. 2.4: Трансляция, компоновка, запуск измененного исполняемого файла lab 7-1

В результате исполнения программы выводится символ с кодом 10. По таблице ASCII этот код соответствует символу перевода строки (LF, `\n`). Сам символ не отображается, но перевод строки был осуществлен.

4. Создаю файл `lab7-2.asm` в том же каталоге, введя `touch ~/work/arch-pc/lab07/lab7-2.asm` и ввожу в него текст программы из листинга 7.2, сохраняю и открываю файл для просмотра (рис. 2.5).

A screenshot of a code editor window titled 'mc [ngalacan@fedora]:~/work/arch-pc/lab07'. The editor shows the file `/home/ngalacan/work/arch-pc/lab07/lab7-2.asm` with 120/120 lines and 100% zoom. The code is as follows:

```
%include 'in_out.asm'

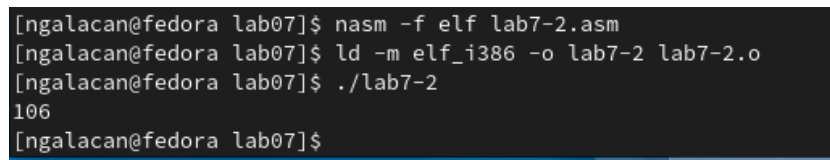
SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 2.5: Текст программы lab 7-2

Создаю исполняемый файл и запускаю его (рис. 2.6).

A screenshot of a terminal window showing the following commands and output:

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

Рис. 2.6: Трансляция, компоновка, запуск исполняемого файла lab 7-2

В результате работы программы выводится число 106, то есть код символа *j* благодаря функции `iprintLF`.

5. Снова изменяю текст программы и вместо символов записываю в регистры числа, заменив строки

```
mov eax, '6'
mov ebx, '4'
```

на

```
mov eax,6
mov ebx,4
```

Создаю исполняемый файл и запускаю его

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

Рис. 2.7: Трансляция, компоновка, запуск измененного исполняемого файла lab7-2

В результате работы программы выводится число 10, что и требовалось получить изначально.

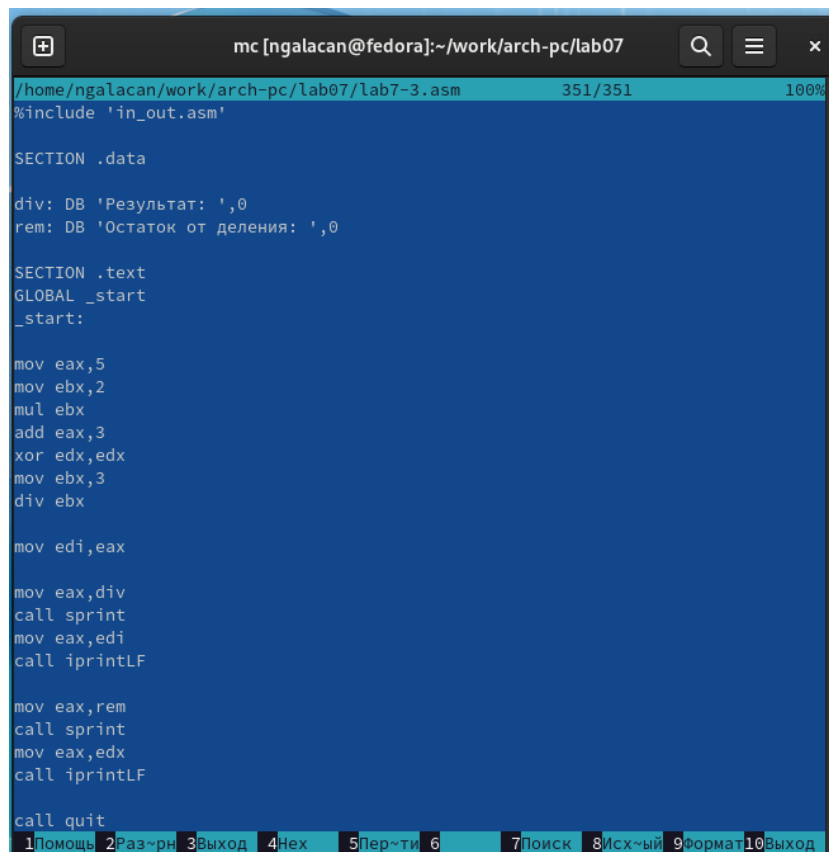
Заменяю функцию `iprintLF` на `iprint`. Создаю исполняемый файл и запускаю его (рис. 2.8).

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

Рис. 2.8: Результат замены функции `iprintLF` на `iprint` в lab7-2

Вывод функции `iprintLF` отличается от `iprint` тем, что в первом случае выводится перевод строки после вывода результата, а во втором нет (аналогично работают `sprintf` и `printf`).

6. Создаю файл `lab7-3.asm`, введя `touch ~/work/arch-pc/lab07/lab7-3.asm`. Изучив текст программы из листинга 7.3, ввожу его в созданный файл, сохраняю и открываю для просмотра (рис. 2.9).



```
mc [ngalacan@fedora]:~/work/arch-pc/lab07
/home/ngalacan/work/arch-pc/lab07/lab7-3.asm 351/351 100%
#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

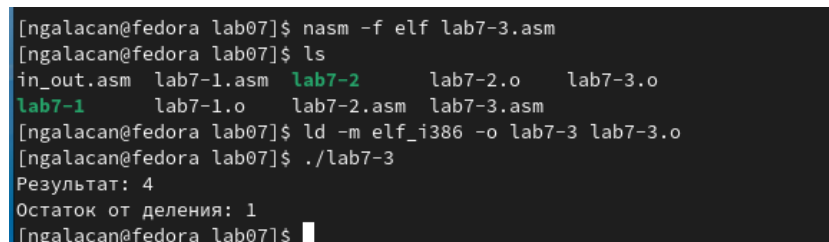
mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
```

Рис. 2.9: Текст программы lab7-3.asm

Создаю исполняемый файл и запускаю его (рис. 2.10).

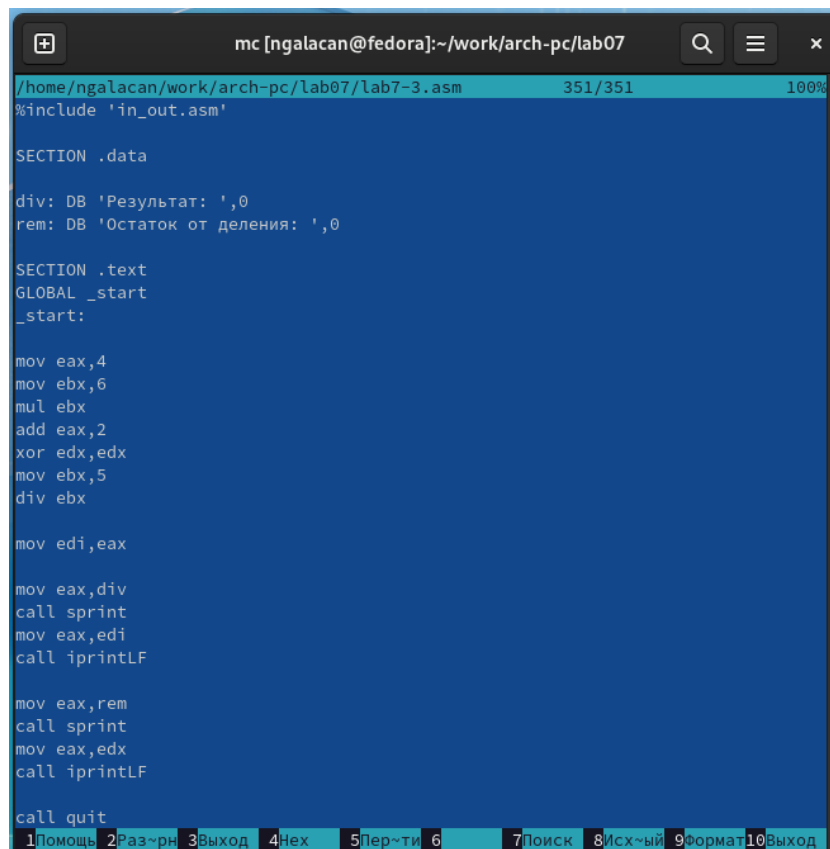


```
[ngalacan@fedora lab07]$ nasm -f elf lab7-3.asm
[ngalacan@fedora lab07]$ ls
in_out.asm lab7-1.asm lab7-2 lab7-2.o lab7-3.o
lab7-1 lab7-1.o lab7-2.asm lab7-3.asm
[ngalacan@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[ngalacan@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[ngalacan@fedora lab07]$
```

Рис. 2.10: Трансляция, компоновка, запуск исполняемого файла lab7-3

Программа вычисляет значение выражения $(5*2+3)/3$, выводит результат и остаток от деления.

Изменяю текст программы для вычисления выражения $(4*6+2)/5$ (рис. 2.11).
Создаю исполняемый файл и запускаю (рис. 2.12).



```
mc [ngalacan@fedora]:~/work/arch-pc/lab07
/home/ngalacan/work/arch-pc/lab07/lab7-3.asm 351/351 100%
#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

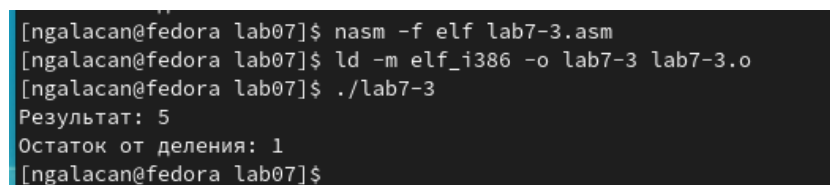
mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
1Помощь 2Разрыв 3Выход 4Чех 5Перти 6 7Поиск 8Исх-ый 9Формат10Выход
```

Рис. 2.11: Измененный текст программы lab7-3.asm

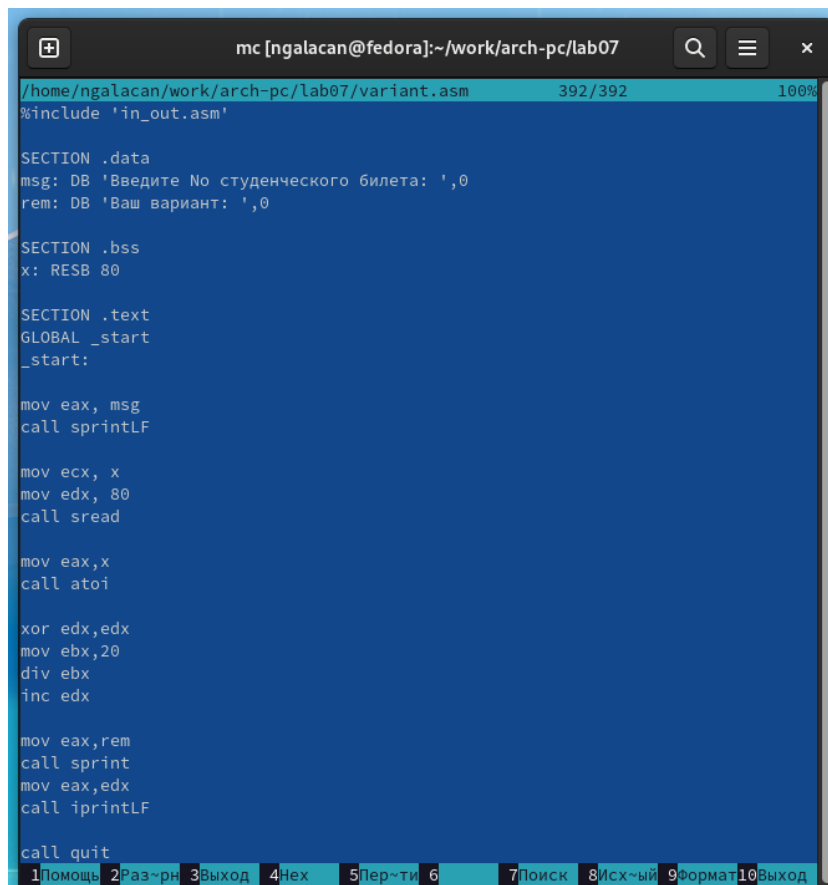


```
[ngalacan@fedora lab07]$ nasm -f elf lab7-3.asm
[ngalacan@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[ngalacan@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[ngalacan@fedora lab07]$
```

Рис. 2.12: Трансляция, компоновка, запуск измененного исполняемого файла lab7-3

В результате работы программа выводит верный ответ и остаток от деления.

7. Создаю файл variant.asm, введя touch ~/work/arch-pc/lab07/variant.asm. Ввожу в него текст программы из листинга 7.4, предварительно изучив его (рис. 2.13).



```
mc [ngalacan@fedora]:~/work/arch-pc/lab07
/home/ngalacan/work/arch-pc/lab07/variant.asm 392/392 100%
#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

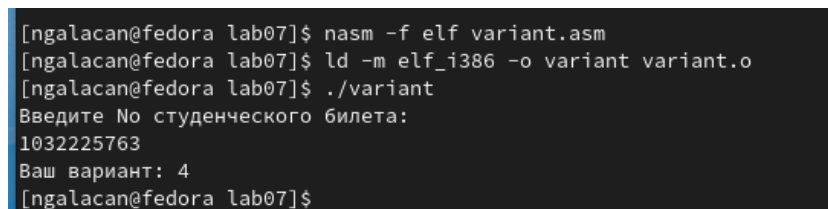
xor edx, edx
mov ebx, 20
div ebx
inc edx

mov eax, rem
call sprintf
mov eax, edx
call iprintLF

call quit
```

Рис. 2.13: Текст программы variant.asm

Создаю исполняемый файл и запускаю (рис. 2.14).



```
[ngalacan@fedora lab07]$ nasm -f elf variant.asm
[ngalacan@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[ngalacan@fedora lab07]$ ./variant
Введите No студенческого билета:
1032225763
Ваш вариант: 4
[ngalacan@fedora lab07]$
```

Рис. 2.14: Трансляция, компоновка, запуск исполняемого файла variant

В результате работы программа выводит номер моего варианта (4) , который совпадает с вычисленным аналитически.

Ответы на вопросы:

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

```
mov eax,rem  
call sprint
```

2. Для чего используются следующие инструкции *asm*?

```
mov ecx, x  
mov edx, 80  
call sread
```

Для считывания переменной *x*, необходимой для вычисления варианта.

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

Инструкция используется для преобразования символа, введенного с клавиатуры, в число для корректного вычисления выражения.

4. Какие строки листинга 7.4 отвечают за вычисления варианта?

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

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

В регистр *edx*.

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

Для прибавления 1 к операнду (*edx*).

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

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

3 Выполнение заданий для самостоятельной работы

В соответствии с полученным вариантом (4), мне необходимо написать программу для вычисления выражения $f(x)=4/3(x-1)+5$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат вычислений.

Создаю файл `lab7-sam.asm` и набираю текст программы. Написанная программа имеет следующий вид:

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
variant: DB 'Вариант 4. Выражение: 4/3(x-1)+5.',0
```

```
vvod: DB 'Введите x: ',0
```

```
rez: DB 'Результат: ',0
```

```
SECTION .bss
```

```
x: RESB 80
```

```
SECTION .text
```

```
GLOBAL _start
```

```
_start:
```

```
mov eax, variant
call sprintf
```

```
mov eax, vvod
call sprintf
```

```
mov ecx, x
mov edx, 80
call sread
```

```
mov eax, x
call atoi
```

```
dec eax ; x-1
```

```
mov ebx, 4 ; (x-1)*4
mul ebx
```

```
xor edx,edx ;(x-1)*4/3
mov ebx, 3
div ebx
```

```
add eax, 5 ;(x-1)*4/3+5
```

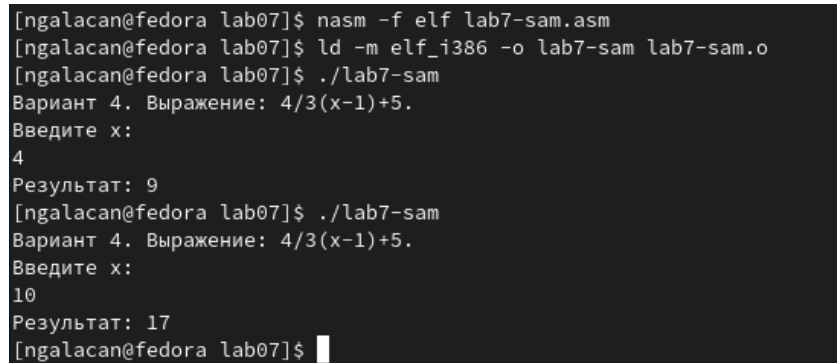
```
mov edi,eax
```

```
mov eax, rez
call sprintf
```

```
mov eax, edi
call iprintLF
```

```
call quit
```

Создаю исполняемый файл и запускаю, ввожу значения x , заданные в таблице 7.3 (рис. 3.1).



```
[ngalacan@fedora lab07]$ nasm -f elf lab7-sam.asm
[ngalacan@fedora lab07]$ ld -m elf_i386 -o lab7-sam lab7-sam.o
[ngalacan@fedora lab07]$ ./lab7-sam
Вариант 4. Выражение: 4/3(x-1)+5.
Введите x:
4
Результат: 9
[ngalacan@fedora lab07]$ ./lab7-sam
Вариант 4. Выражение: 4/3(x-1)+5.
Введите x:
10
Результат: 17
[ngalacan@fedora lab07]$
```

Рис. 3.1: Трансляция, компоновка, запуск исполняемого файла lab7-sam

Программа работает в соответствии с заданием: выводит выражение, запрашивает x , вычисляет выражение и выводит верный результат.

4 Выводы

На данной лабораторной работе были освоены арифметические инструкции языка ассемблера NASM и их особенности.