

Отчет по лабораторной работе №7

дисциплина: Архитектура компьютера

Колобова Елизавета Андреевна гр. НММбд-01

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Задание для самостоятельной работы	17
4	Выводы	20

Список иллюстраций

3.1	Рис. 1. Создание каталога для лаб. работы №7 и файла lab7-1.asm .	7
3.2	Рис. 2. Ввод текста из листинга 7.1	8
3.3	Рис. 3. Компоновка и запуск файла lab7-1	8
3.4	Рис. 4. Изменение текста программы	9
3.5	Рис. 5. Компоновка и запуск измененного файла	9
3.6	Рис. 6. Ввод текста из листинга 7.2	10
3.7	Рис. 7. Компоновка и запуск файла	10
3.8	Рис. 8. Окно текстового редактора	11
3.9	Рис. 9. Окно текстового редактора	11
3.10	Рис. 10. Замена функции <code>iprintLF</code> на <code>iprint</code> в тексте программы . .	12
3.11	Рис. 11. Запуск измененной программы	12
3.12	Рис. 12. Ввод текста программы из листинга 7.3	13
3.13	Рис. 13. Ввод текста программы из листинга 7.3	13
3.14	Рис. 14. Компоновка и запуск файла	13
3.15	Рис. 15. Изменение текста программы	14
3.16	Рис. 16. Компоновка и запуск файла с изменениями	14
3.17	Рис. 17. Ввод текста программы из листинга 7.3	15
3.18	Рис. 18. Ввод текста программы из листинга 7.3	16
3.19	Рис. 19. Компоновка и запуск файла	16
3.20	Рис. 20. Текст программы	18
3.21	Рис. 21. Текст программы	18
3.22	Рис. 22. Компоновка и запуск файла	19

Список таблиц

1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

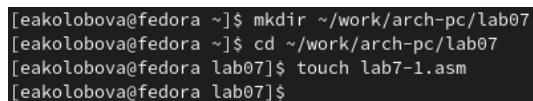
2 Задание

Написать на языке ассемблера программы вывода значения регистра и вычисления выражения.

3 Выполнение лабораторной работы

1. Создаем каталог для программ лабораторной работы No 7, переходим в него и создаем файл lab7-1.asm (рис. 3.1):

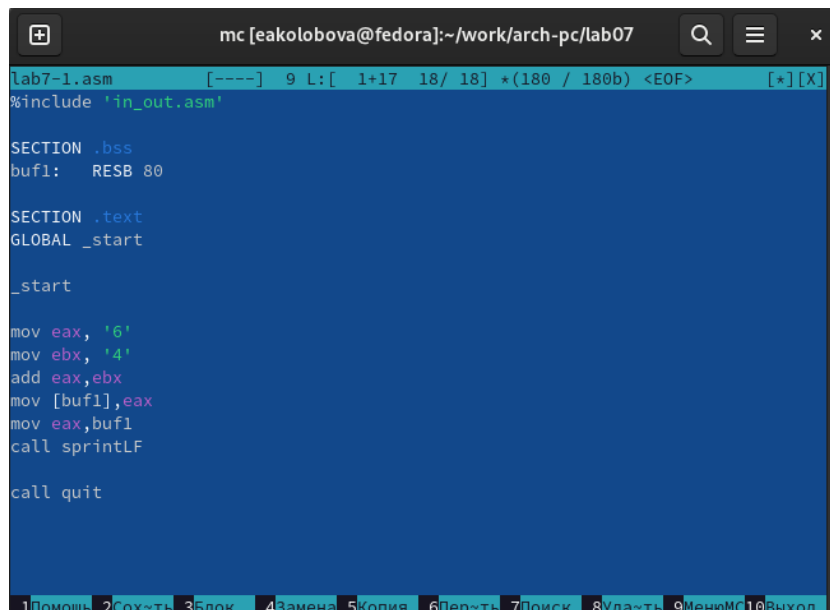
```
mkdir ~/work/arch-pc/lab07  
cd ~/work/arch-pc/lab07  
touch lab7-1.asm
```



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

Рис. 3.1: Рис. 1. Создание каталога для лаб. работы №7 и файла lab7-1.asm

2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. Введем в файл lab7-1.asm текст программы из листинга 7.1. (рис. 3.2).



```
lab7-1.asm [----] 9 L: [ 1+17 18/ 18] *(180 / 180b) <EOF> [*][X]
#include 'in_out.asm'

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start

_start

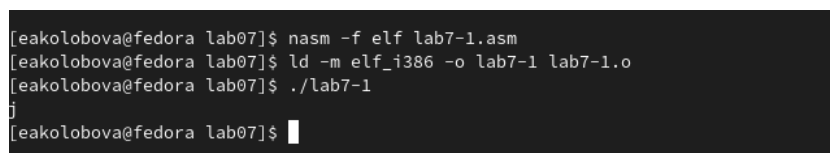
mov eax, '6'
mov ebx, '4'
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintLF

call quit
```

Рис. 3.2: Рис. 2. Ввод текста из листинга 7.1

3. Создаем исполняемый файл и запускаем его. (рис. 3.3)

```
nasm -f elf lab7-1.asm
ld -m elf_i386 -o lab7-1 lab7-1.o
./lab7-1 (рис. [-@fig:003])
```



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

Рис. 3.3: Рис. 3. Компоновка и запуск файла lab7-1

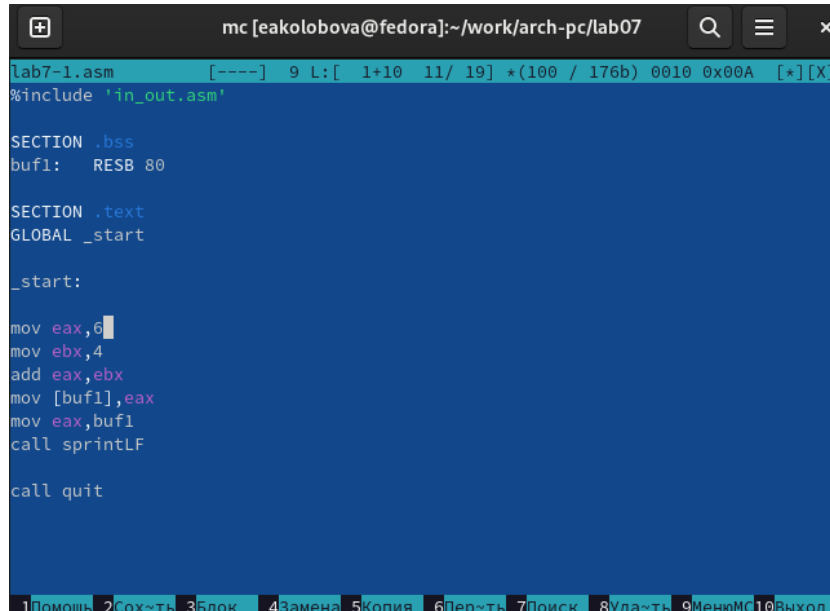
4. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы (Листинг 1) следующим образом (рис. 3.4): заменим строки

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

на строки


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

Создаем исполняемый файл и запускаем его. (рис. 3.5)



```
lab7-1.asm [----] 9 L: [ 1+10 11/ 19] *(100 / 176b) 0010 0x00A [*][X]
#include 'in_out.asm'

SECTION .bss
buf1:  RESB 80

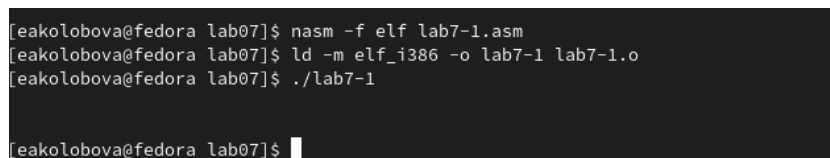
SECTION .text
GLOBAL _start

_start:

mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF

call quit
```

Рис. 3.4: Рис. 4. Изменение текста программы



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

[eakolobova@fedora lab07]$
```

Рис. 3.5: Рис. 5. Компоновка и запуск измененного файла

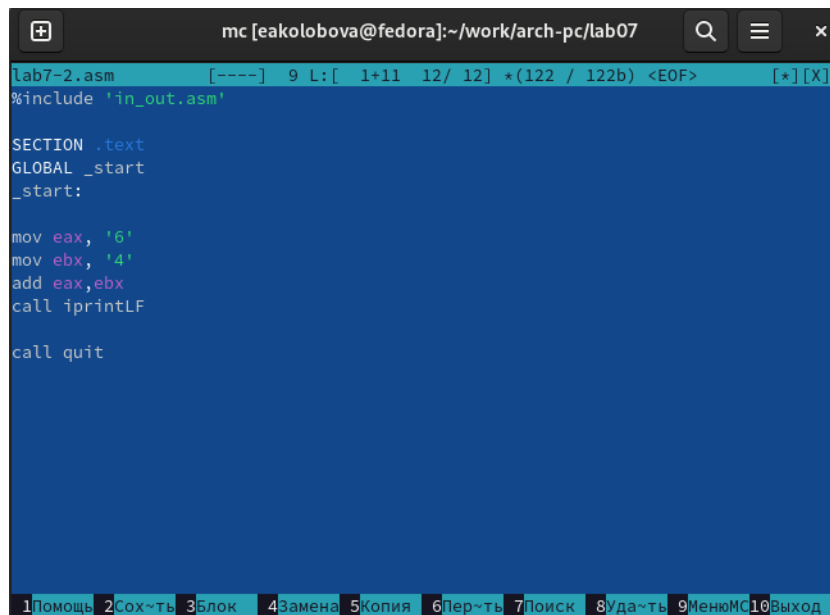
5. Создаем файл lab7-2.asm в каталоге ~/work/arch-pc/lab07

```
touch ~/work/arch-pc/lab07/lab7-2.asm.
```

и вводим в него текст программы из листинга 7.2. Создаем исполняемый файл и запускаем его.

```
nasm -f elf lab7-1.asm
ld -m elf_i386 -o lab7-1 lab7-1.o
./lab7-1
```

(рис. 3.6, 3.7)



The screenshot shows a text editor window titled 'mc [eakolobova@fedora]:~/work/arch-pc/lab07'. The editor contains the following assembly code:

```
lab7-2.asm  [----]  9  L: [ 1+11 12/ 12] *(122 / 122b) <EOF>  [*][X]
#include 'in_out.asm'

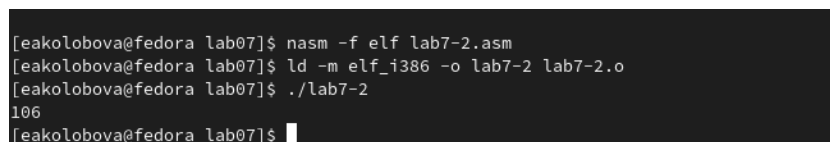
SECTION .text
GLOBAL _start
_start:

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

call quit
```

At the bottom of the window, there is a menu bar with the following items: 1Помощь, 2Сох~ть, 3Блок, 4Замена, 5Копия, 6Пер~ть, 7Поиск, 8Уда~ть, 9Меню, 10Выход.

Рис. 3.6: Рис. 6. Ввод текста из листинга 7.2



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

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

Рис. 3.7: Рис. 7. Компоновка и запуск файла

6. Заменим строки

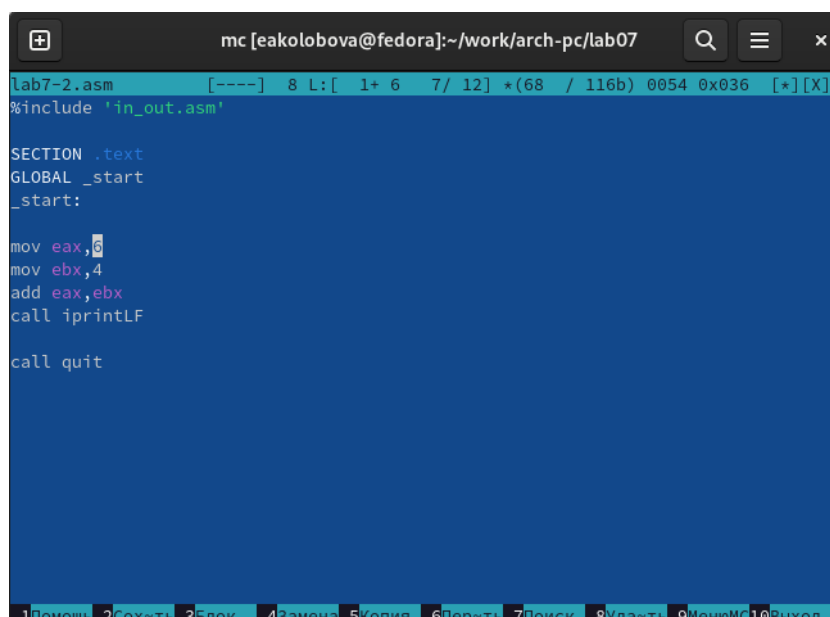
```
mov eax, '6'
mov ebx, '4'
```

на строки

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

Создаем исполняемый файл и запускаем его. (рис. 3.8, 3.9) При исполнении программы получен результат 10, т.е. результат сложения 6+4

Заменим функцию `iprintLF` на `iprint`. Создаем исполняемый файл и запускаем его. Вывод функций `iprintLF` отличается от `iprint` наличием перевода строки. (рис. 3.10, 3.11)



The screenshot shows a text editor window titled "mc [eakolobova@fedora]:~/work/arch-pc/lab07". The editor contains the following assembly code:

```
lab7-2.asm [----] 8 L: [ 1+ 6 7/ 12] *(68 / 116b) 0054 0x036 [*][X]
#include 'in_out.asm'

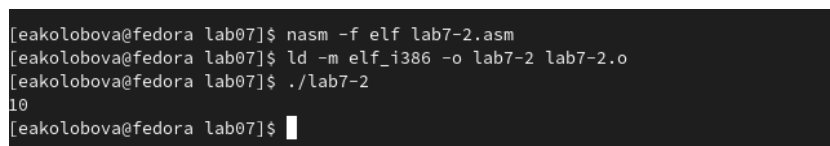
SECTION .text
GLOBAL _start
_start:

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

call quit
```

At the bottom of the window, there is a menu bar with the following items: 1.Помощь, 2.Сох-ть, 3.Блок, 4.Замена, 5.Копия, 6.Пер-ть, 7.Поиск, 8.Уда-ть, 9.МенюМС, 10.Выход.

Рис. 3.8: Рис. 8. Окно текстового редактора



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

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

Рис. 3.9: Рис. 9. Окно текстового редактора

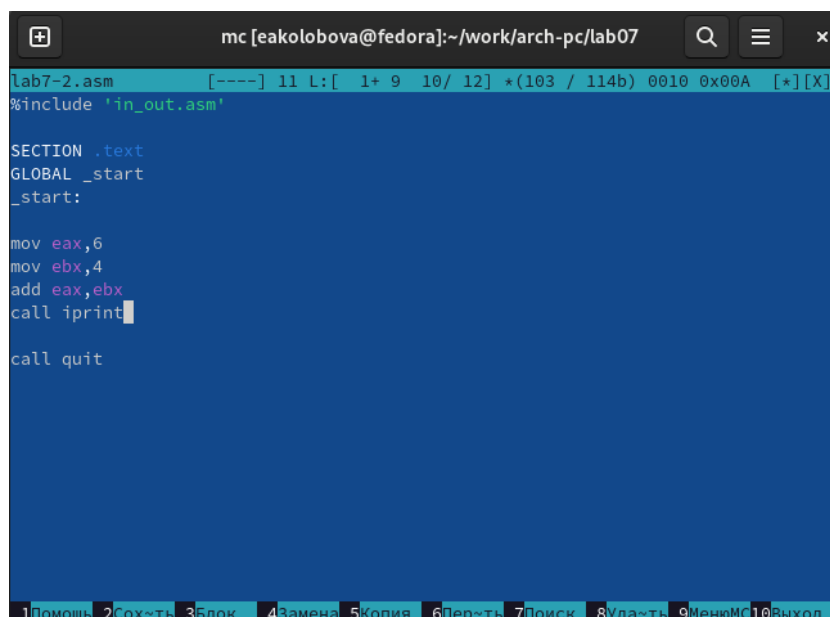


Рис. 3.10: Рис. 10. Замена функции iprintLF на iprint в тексте программы

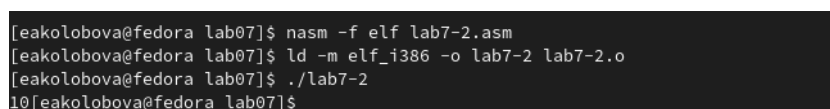


Рис. 3.11: Рис. 11. Запуск измененной программы

7. Рассмотрим программу вычисления арифметического выражения $f(x) = (5 * 2 + 3)/3$.

Создаем файл lab7-3.asm в каталоге ~/work/arch-pc/lab07:

`touch ~/work/arch-pc/lab07/lab7-3.asm`

Вводим текст программы из листинга 7.3 в lab7-3.asm Создаем исполняемый файл и запускаем его. Результат работы программы должен быть следующим:

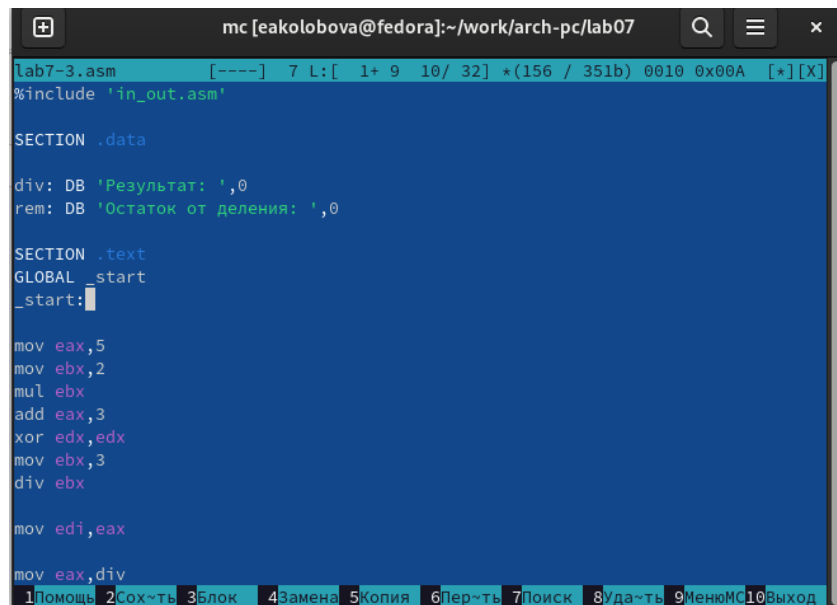
```
user@dk4n31:~$ ./lab7-3
```

```
Результат: 4
```

```
Остаток от деления: 1
```

```
user@dk4n31:~$
```

Как видно по рис. 3.14, полученный результат совпадает с требуемым.(рис. 3.12, 3.13, 3.14)



```
lab7-3.asm [----] 7 L: [ 1+ 9 10/ 32] *(156 / 351b) 0010 0x00A [*][X]
#include 'in_out.asm'

SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

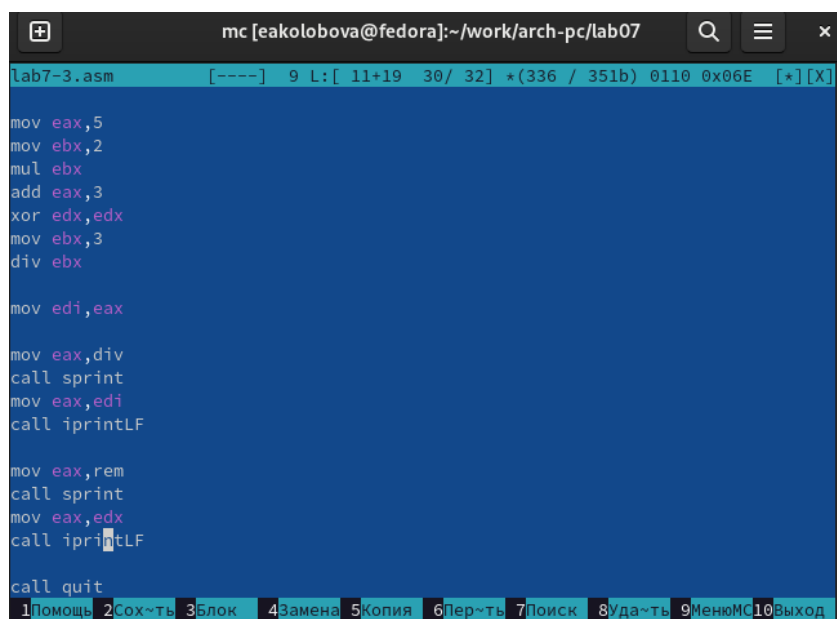
mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

mov eax,div

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9Меню 10Выход
```

Рис. 3.12: Рис. 12. Ввод текста программы из листинга 7.3



```
lab7-3.asm [----] 9 L: [ 11+19 30/ 32] *(336 / 351b) 0110 0x06E [*][X]

mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

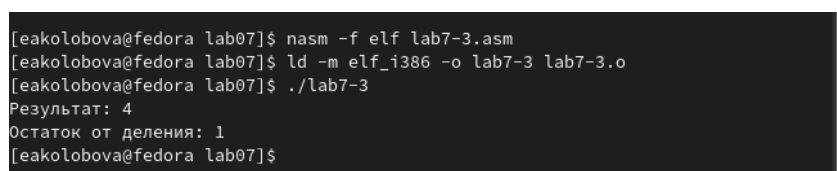
mov eax,div
call sprint
mov eax,edi
call iprintLF

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

call quit

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9Меню 10Выход
```

Рис. 3.13: Рис. 13. Ввод текста программы из листинга 7.3



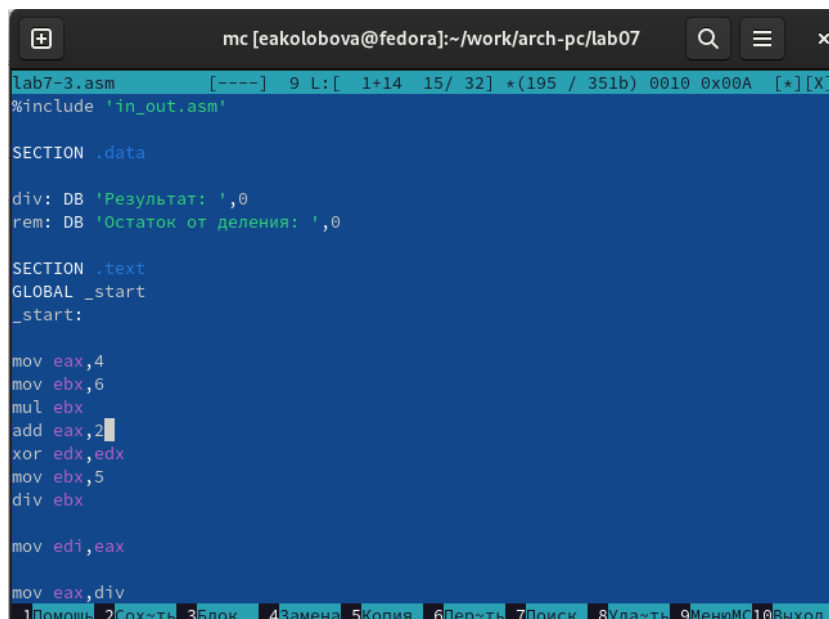
```
[eakolobova@fedora lab07]$ nasm -f elf lab7-3.asm
[eakolobova@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[eakolobova@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[eakolobova@fedora lab07]$
```

Рис. 3.14: Рис. 14. Компоновка и запуск файла

8. Изменим текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$.

Создаем исполняемый файл и проверяем его работу. (рис. 3.15, 3.16)

```
user@dk4n31:~$ nasm -f elf lab6-1.asm
user@dk4n31:~$ ld -m elf_i386 -o lab6-1 lab6-1.o
user@dk4n31:~$ ./lab6-1
Введите строку:
Имя пользователя
user@dk4n31:~$
```



```
lab7-3.asm  [----]  9  L: [ 1+14 15/ 32] *(195 / 351b) 0010 0x00A  [*][X]
#include 'in_out.asm'

SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

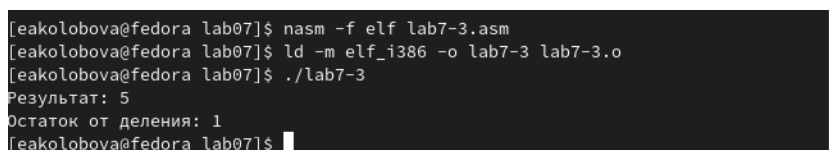
SECTION .text
GLOBAL _start
_start:

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
```

Рис. 3.15: Рис. 15. Изменение текста программы

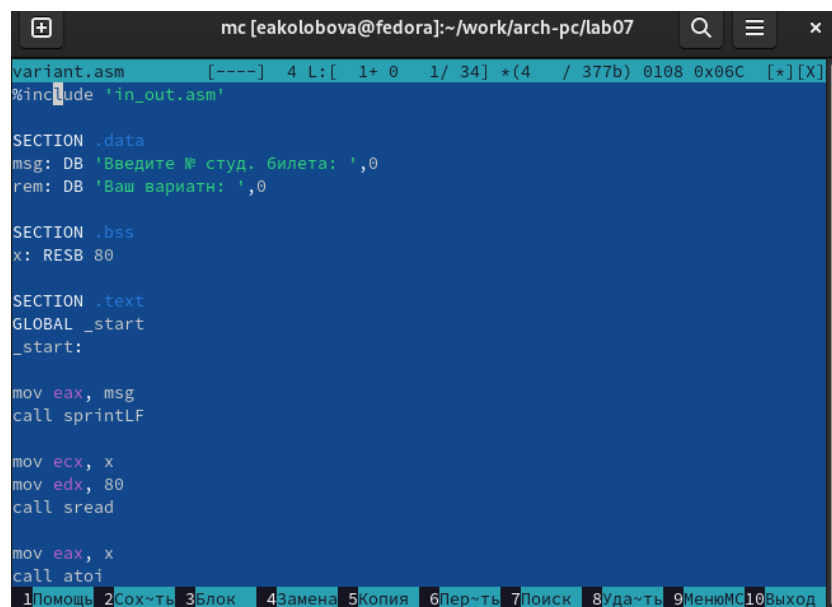


```
[eakolobova@fedora lab07]$ nasm -f elf lab7-3.asm
[eakolobova@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[eakolobova@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[eakolobova@fedora lab07]$
```

Рис. 3.16: Рис. 16. Компоновка и запуск файла с изменениями

9. Рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:

10. вывести запрос на введение студенческого билета
11. вычислить номер варианта по формуле: $(S_n \bmod 20) + 1$, где S_n – номер студенческого билета (В данном случае $a \bmod b$ – это остаток от деления a на b).
12. вывести на экран номер вариант Создаем файл variant.asm в каталоге `~/work/arch-pc/lab07`: `touch ~/work/arch-pc/lab07/variant.asm` Вводим текст программы из листинга 7.4 в файл variant.asm. Создаем исполняемый файл и запускаем его. Проверяем результат работы. (рис. 3.17, 3.18, 3.19)



```
mc [eakolobova@fedora]:~/work/arch-pc/lab07
variant.asm [----] 4 L: [ 1+ 0 1/ 34] *(4 / 377b) 0108 0x06C [*][X]
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите № студ. билета: ',0
rem: DB 'Ваш вариант: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

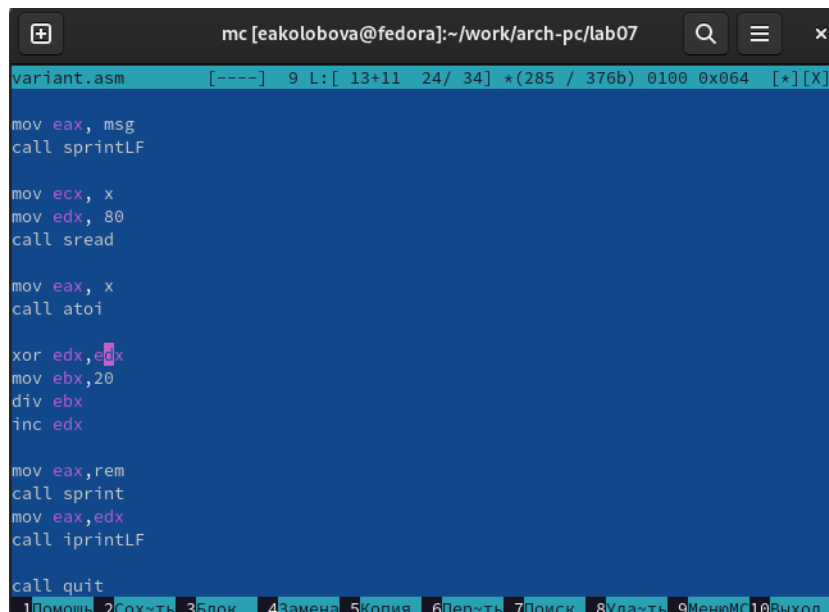
mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

1Помощь 2Сох~ть 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюМС10Выход
```

Рис. 3.17: Рис. 17. Ввод текста программы из листинга 7.3



```
variant.asm  [----]  9  L:[ 13+11  24/ 34]  *(285 / 376b) 0100 0x064  [*][X]

mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
call sread

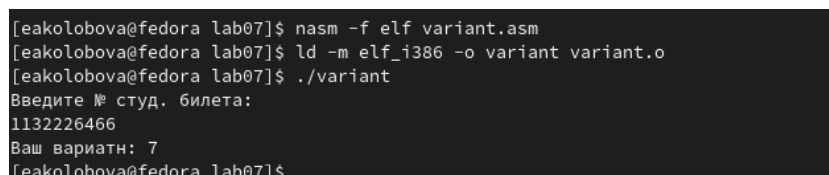
mov eax, x
call atoi

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

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

call quit
```

Рис. 3.18: Рис. 18. Ввод текста программы из листинга 7.3



```
[eakolobova@fedora lab07]$ nasm -f elf variant.asm
[eakolobova@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[eakolobova@fedora lab07]$ ./variant
Введите № студ. билета:
1132226466
Ваш вариант: 7
[eakolobova@fedora lab07]$
```

Рис. 3.19: Рис. 19. Компоновка и запуск файла

10. Вопросы:

11. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? -

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

2. Для чего используются следующие инструкции? `mov ecx, x`, `mov edx, 80`, `call sread` - для считывания введенного с клавиатуры значения `x`

3. Для чего используется инструкция “`call atoi`”? - для преобразования ASCII кода в число, `eax=x`

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


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

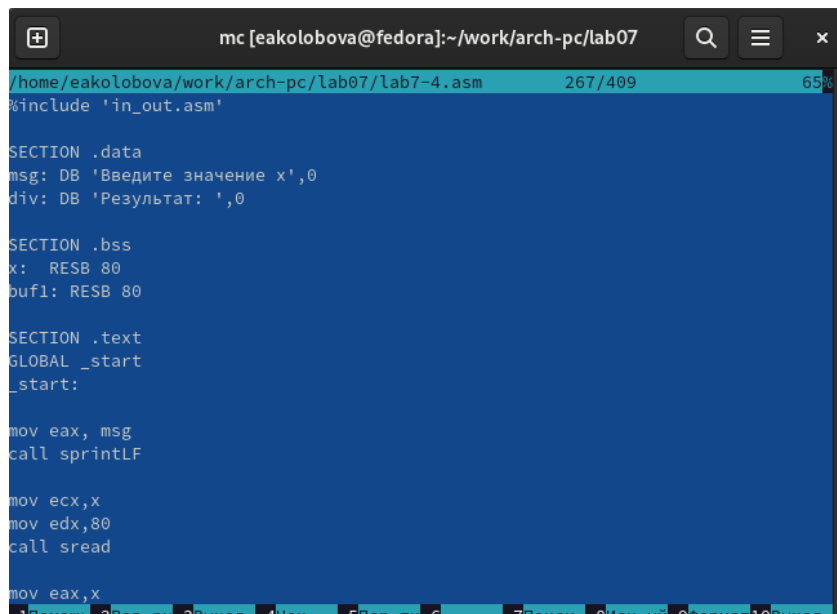
5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? - в edx
6. Для чего используется инструкция “inc edx”? - для увеличения значения регистра edx на 1
7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений? -

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

3.1 Задание для самостоятельной работы

1. Написать программу вычисления выражения $y = f(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат вычислений. Вид функции $f(x)$ выбрать из таблицы 7.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 из 7.3

Из программы (рис. 3.19), получаем, что номеру студ.билета соответствует вариант 7. Выражение: $5 \cdot (x - 1)^2$, $x_1=3$, $x_2=5$. Для получения нужной программы вводим текст, представленный на рис. 3.20, 3.21. Создаем исполняемый файл и запускаем его. Проверяем результат работы на значениях x_1 и x_2 . (рис. 3.22)



```
mc [eakolobova@fedora]:~/work/arch-pc/lab07
/home/eakolobova/work/arch-pc/lab07/lab7-4.asm 267/409 65%
#include 'in_out.asm'

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

SECTION .bss
x: RESB 80
buf1: RESB 80

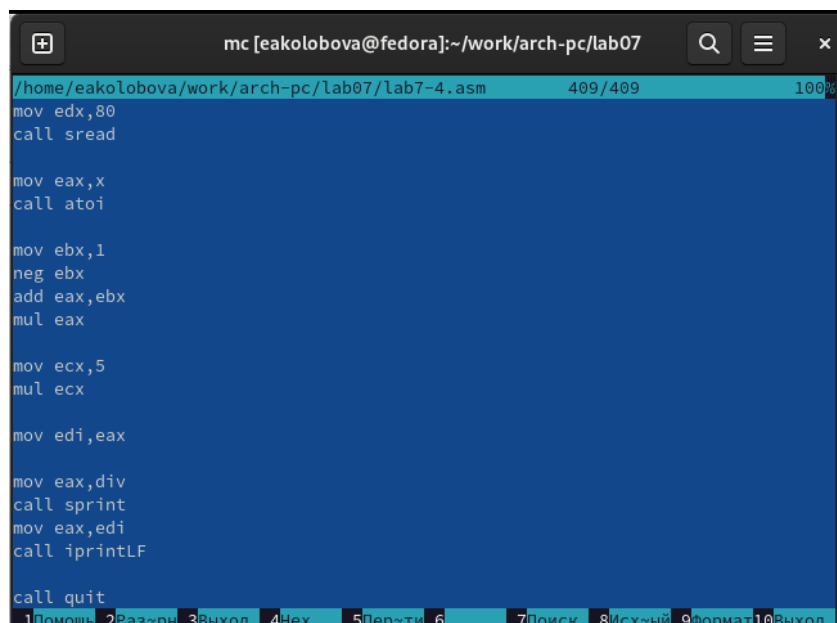
SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
call sread

mov eax, x
```

Рис. 3.20: Рис. 20. Текст программы



```
mc [eakolobova@fedora]:~/work/arch-pc/lab07
/home/eakolobova/work/arch-pc/lab07/lab7-4.asm 409/409 100%
mov edx, 80
call sread

mov eax, x
call atoi

mov ebx, 1
neg ebx
add eax, ebx
mul eax

mov ecx, 5
mul ecx

mov edi, eax

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

call quit
```

Рис. 3.21: Рис. 21. Текст программы

```
[eakolobova@fedora lab07]$ nasm -f elf lab7-4.asm
[eakolobova@fedora lab07]$ ld -m elf_i386 -o lab7-4 lab7-4.o
[eakolobova@fedora lab07]$ ./lab7-4
Введите значение x
3
Результат: 20
[eakolobova@fedora lab07]$ ./lab7-4
Введите значение x
5
Результат: 80
[eakolobova@fedora lab07]$
```

Рис. 3.22: Рис. 22. Компоновка и запуск файла

Ссылка на репозиторий: https://github.com/eakolobova/study_2022-2023_arch-pc/tree/master/labs/lab07/report

4 Выводы

Результатом проведенной работы является освоение арифметических инструкций языка ассемблера NASM.