

# **Лабораторная работа №7**

**Арифметические операции в NASM**

Медникова Екатерина Михайловна

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Ответы на вопросы	15
5	Самостоятельная работа	16
6	Выводы	20

# Список иллюстраций

3.1	Создание каталога . . . . .	7
3.2	Переход в каталог . . . . .	7
3.3	Создание файла . . . . .	8
3.4	Ввод текста . . . . .	8
3.5	Результат . . . . .	8
3.6	Изменение текста . . . . .	9
3.7	Результат . . . . .	9
3.8	Создание файла . . . . .	9
3.9	Ввод текста . . . . .	10
3.10	Результат . . . . .	10
3.11	Корректировка текста . . . . .	10
3.12	Результат . . . . .	11
3.13	Замена функции . . . . .	11
3.14	Результат . . . . .	11
3.15	Ввод текста программы . . . . .	12
3.16	Результат . . . . .	12
3.17	Изменение текста . . . . .	13
3.18	Проверка работы . . . . .	13
3.19	Создание файла . . . . .	14
3.20	Ввод текста программы . . . . .	14
3.21	Результат . . . . .	14
5.1	Программа . . . . .	17
5.2	Программа . . . . .	18
5.3	Результат . . . . .	19

## Список таблиц

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Теоретическое введение

Микропроцессор может выполнять целочисленные операции и операции с плавающей точкой. Для этого в его архитектуре есть два отдельных блока:

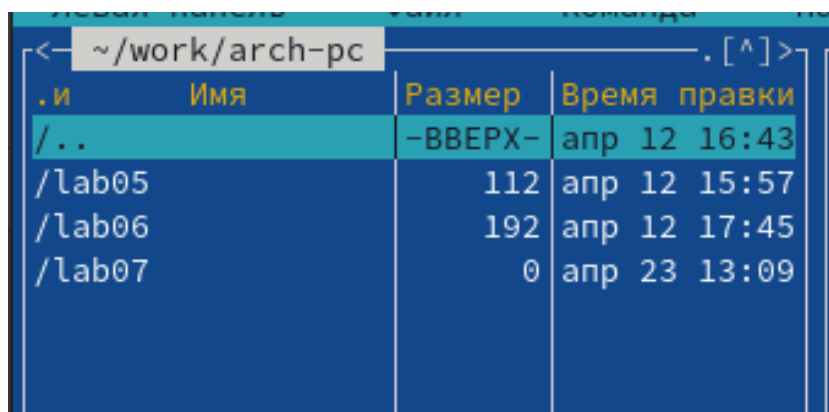
устройство для выполнения целочисленных операций;

устройство с плавающей точкой.

Каждое из этих устройств имеет свою систему команд. Целочисленное устройство может взять на себя многие функции устройства с плавающей точкой, но это потребует больших вычислительных затрат. Для большинства задач, использующих язык ассемблера, достаточно целочисленной арифметики.

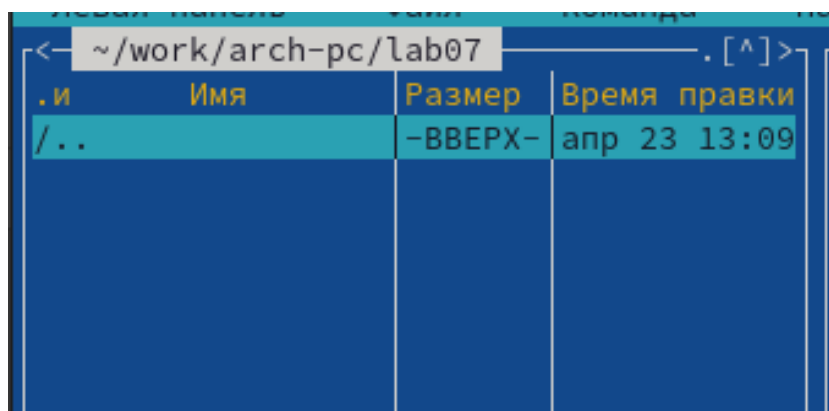
### 3 Выполнение лабораторной работы

1. Создала каталог и файл.



.и	Имя	Размер	Время правки
/..	-ВВЕРХ-		апр 12 16:43
/lab05		112	апр 12 15:57
/lab06		192	апр 12 17:45
/lab07		0	апр 23 13:09

Рис. 3.1: Создание каталога



.и	Имя	Размер	Время правки
/..	-ВВЕРХ-		апр 23 13:09

Рис. 3.2: Переход в каталог

~/work/arch-pc/lab07 .[^]>			
.и	Имя	Размер	Время правки
./..		-ВВЕРХ-	апр 23 13:09
lab7-1.asm		0	апр 23 13:10

Рис. 3.3: Создание файла

2. Ввела в файл текст программы из листинга 7.1. Создала исполняемый файл и запустила его.

```

Открыть ▾ + *lab7-1.asm
~/work/arch-pc/lab07
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 sprintLF
13 call quit

```

Рис. 3.4: Ввод текста

```

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

```

Рис. 3.5: Результат



3. Изменила в файле текст программы, как было указано в лабораторной работе. Создала исполняемый файл и запустила его.

```
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 sprintLF
13 call quit
```

Рис. 3.6: Изменение текста

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

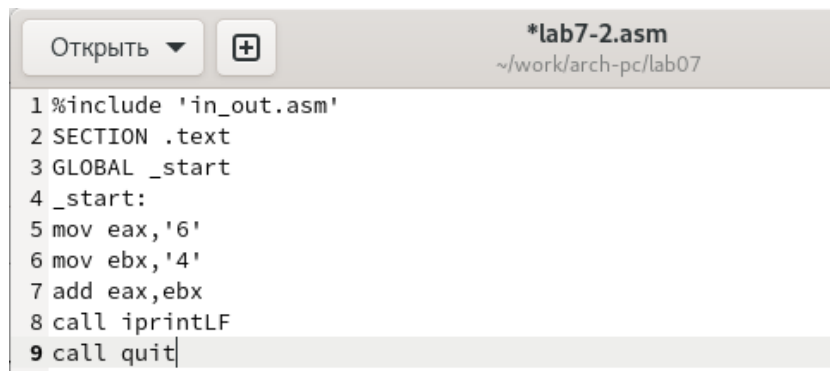
[emmednikova@fedora lab07]$
```

Рис. 3.7: Результат

4. Создала файл lab7-2.asm и ввела в него текст программы из листинга 7.2. Создала исполняемый файл и запустила его.

```
[emmednikova@fedora lab07]$ touch lab7-2.asm
[emmednikova@fedora lab07]$ ls
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o  lab7-2.asm
[emmednikova@fedora lab07]$
```

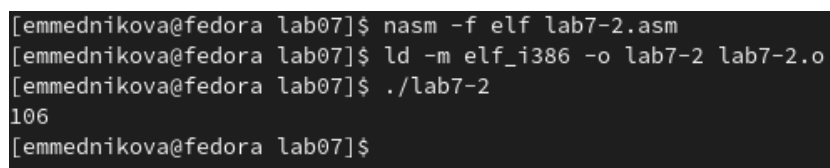
Рис. 3.8: Создание файла



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

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
```

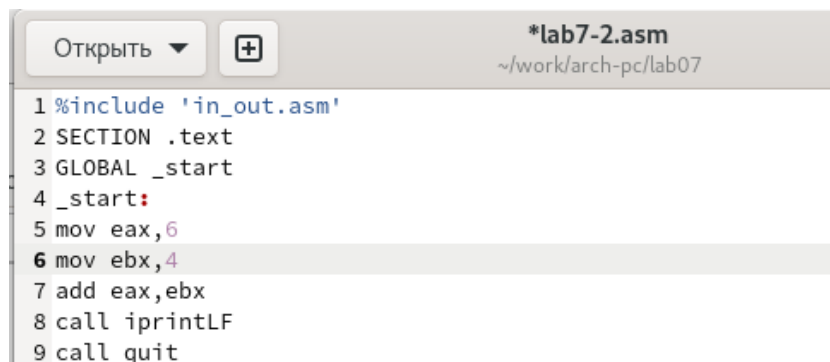
Рис. 3.9: Ввод текста



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

Рис. 3.10: Результат

5. Изменила символы на числа. Создала исполняемый файл и запустила его. При исполнении программы результат получился 10.



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

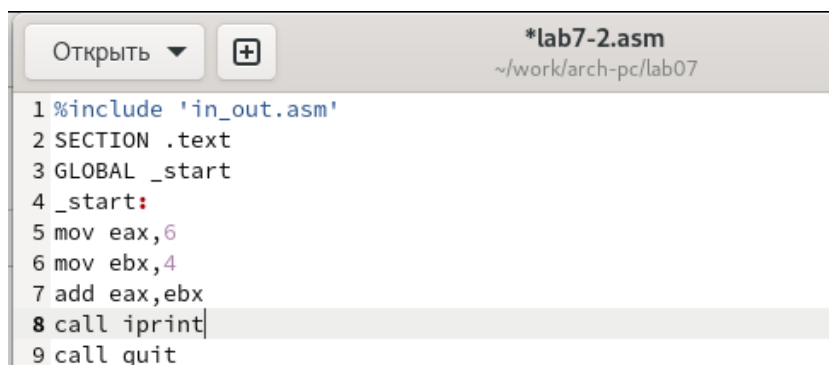
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit
```

Рис. 3.11: Корректировка текста

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

Рис. 3.12: Результат

6. Заменяла функцию `iprintLF` на `iprint`. Создала исполняемый файл и запустила его. При `iprint` числа вводятся в формате ASCII. Функция `iprintLF` работает аналогично, но при выводе на экран после числа добавляет к символу перевод строки.



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

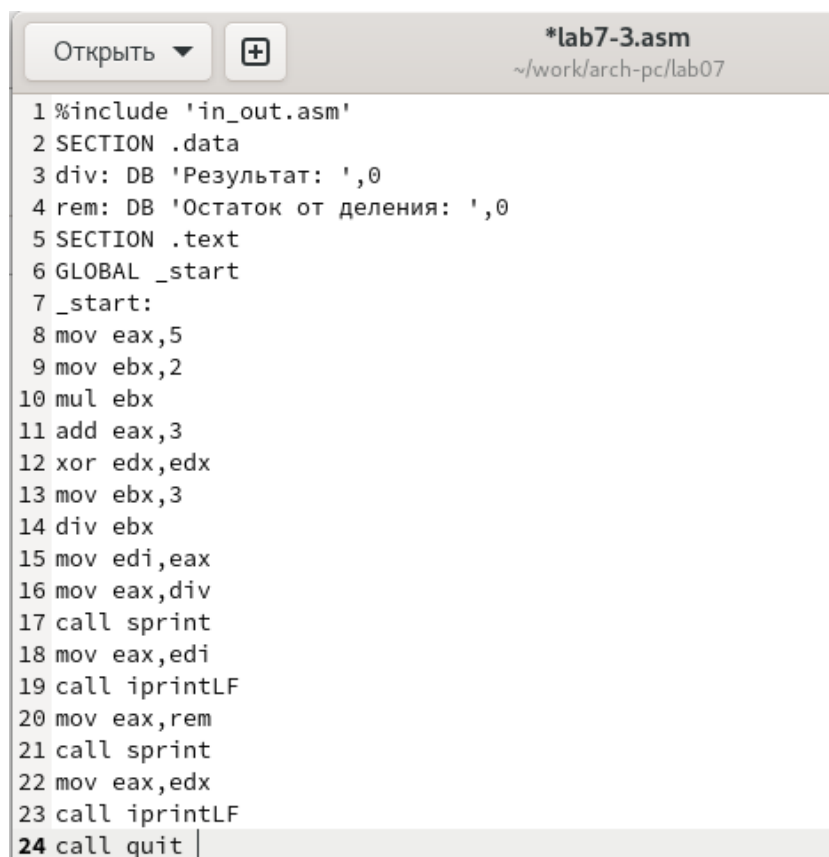
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprint
9 call quit
```

Рис. 3.13: Замена функции

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

Рис. 3.14: Результат

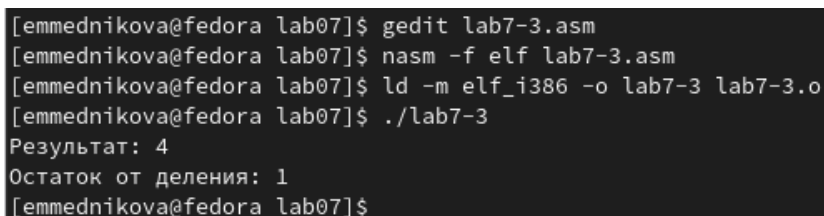
7. Создала файл `lab7-3.asm`. Ввела в него текст программы из листинга 7.3.



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

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,div
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 |
```

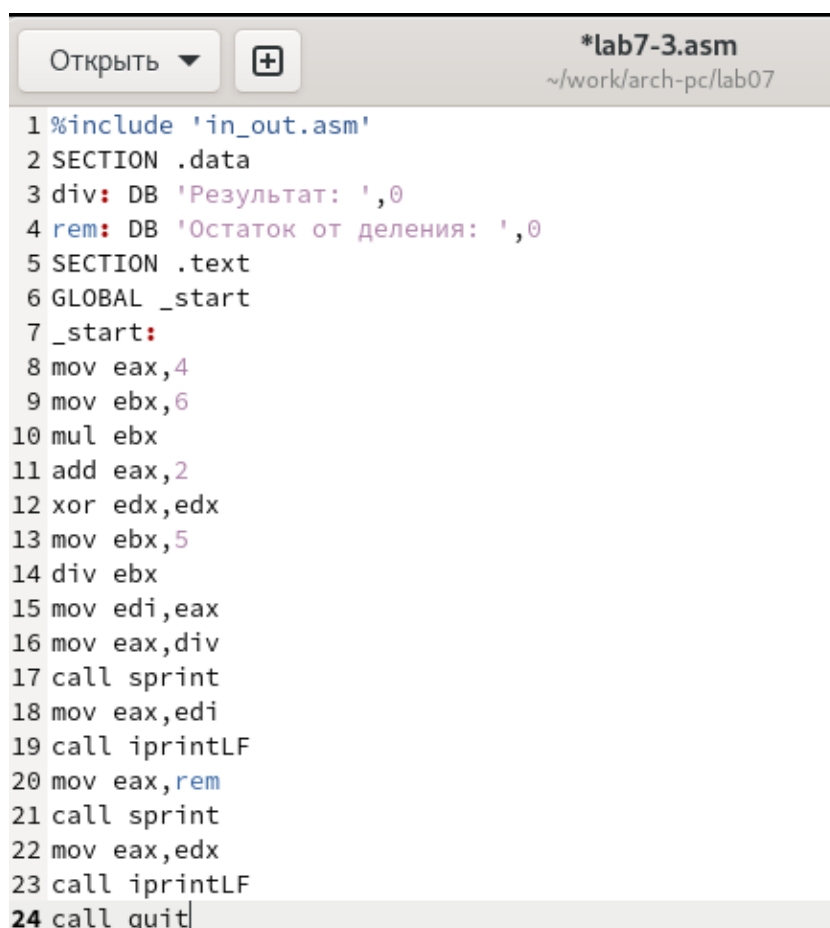
Рис. 3.15: Ввод текста программы



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

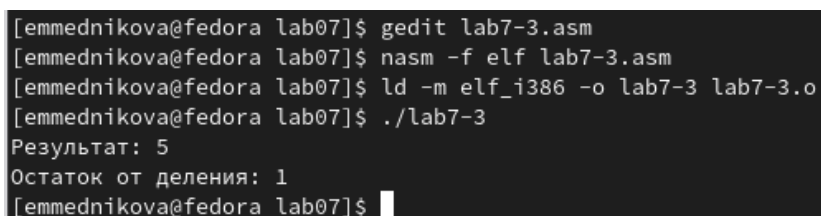
Рис. 3.16: Результат

8. Изменила текст программы для вычисления выражения. Создала исполняемый файл и проверила его работу.



```
Открыть ▾ + *lab7-3.asm ~/work/arch-pc/lab07
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,4
9 mov ebx,6
10 mul ebx
11 add eax,2
12 xor edx,edx
13 mov ebx,5
14 div ebx
15 mov edi,eax
16 mov eax,div
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
```

Рис. 3.17: Изменение текста



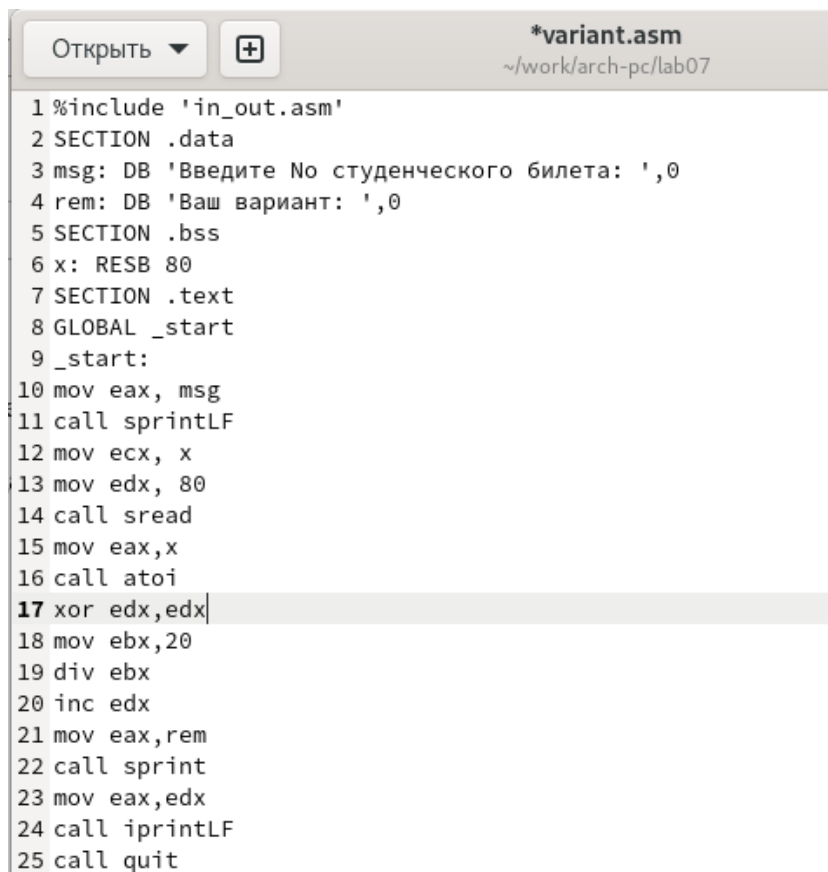
```
[emmednikova@fedora lab07]$ gedit lab7-3.asm
[emmednikova@fedora lab07]$ nasm -f elf lab7-3.asm
[emmednikova@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[emmednikova@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[emmednikova@fedora lab07]$
```

Рис. 3.18: Проверка работы

9. Создала файл variant.asm. Ввела в него текст программы из листинга 7.4.  
Создала исполняемый файл и запустила его.

```
[emmednikova@fedora lab07]$ touch variant.asm
[emmednikova@fedora lab07]$ ls
in_out.asm  lab7-1.asm  lab7-2      lab7-2.o  lab7-3.asm  variant.asm
lab7-1      lab7-1.o    lab7-2.asm  lab7-3    lab7-3.o
[emmednikova@fedora lab07]$
```

Рис. 3.19: Создание файла



```
*variant.asm
~/work/arch-pc/lab07

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите No 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 sprintf
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
```

Рис. 3.20: Ввод текста программы

```
[emmednikova@fedora lab07]$ gedit variant.asm
[emmednikova@fedora lab07]$ nasm -f elf variant.asm
[emmednikova@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[emmednikova@fedora lab07]$ ./variant
Введите No студенческого билета:
1132226549
Ваш вариант: 10
[emmednikova@fedora lab07]$
```

Рис. 3.21: Результат

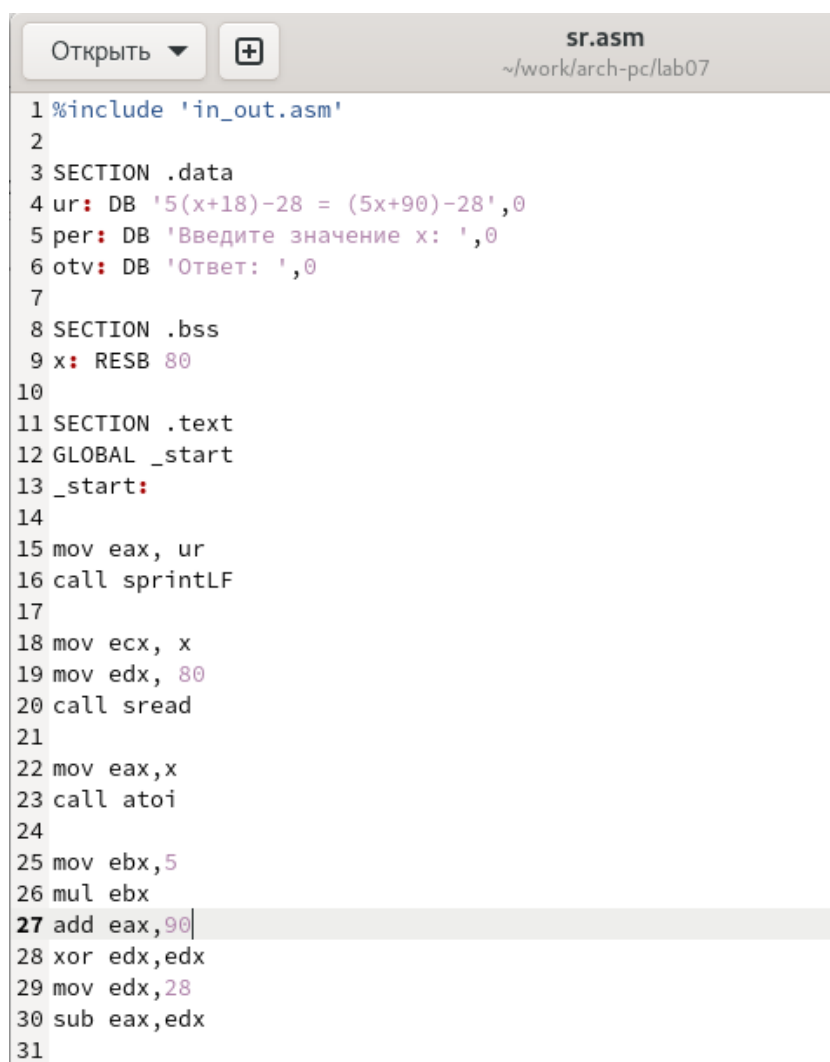
## 4 Ответы на вопросы

1. За вывод на экран сообщения 'Ваш вариант:' отвечают строки `mov eax, ret`  
`call sprint`.
2. `nasm mov ecx` - записывает адрес выводимого сообщения в `ecx`;  
  
`mov edx, 80` - записывает длину вводимого сообщения в `edx`;  
`call sread` - выполняет вызов программы ввода сообщения.
3. Инструкция '`call atoi`' используется для преобразования символа в число.
4. За вычисление варианта отвечают строки: `xor edx, edx` `mov ebx, 20` `div ebx` `inc`  
`edx`.
5. Остаток от деления записывается в регистр `ebx`.
6. Инструкция '`inc edx`' используется для увеличения значения `edx` на единицу.
7. За вывод на экран результата вычислений отвечают строки: `mov eax, edx`  
`call iprintLF`

## 5 Самостоятельная работа

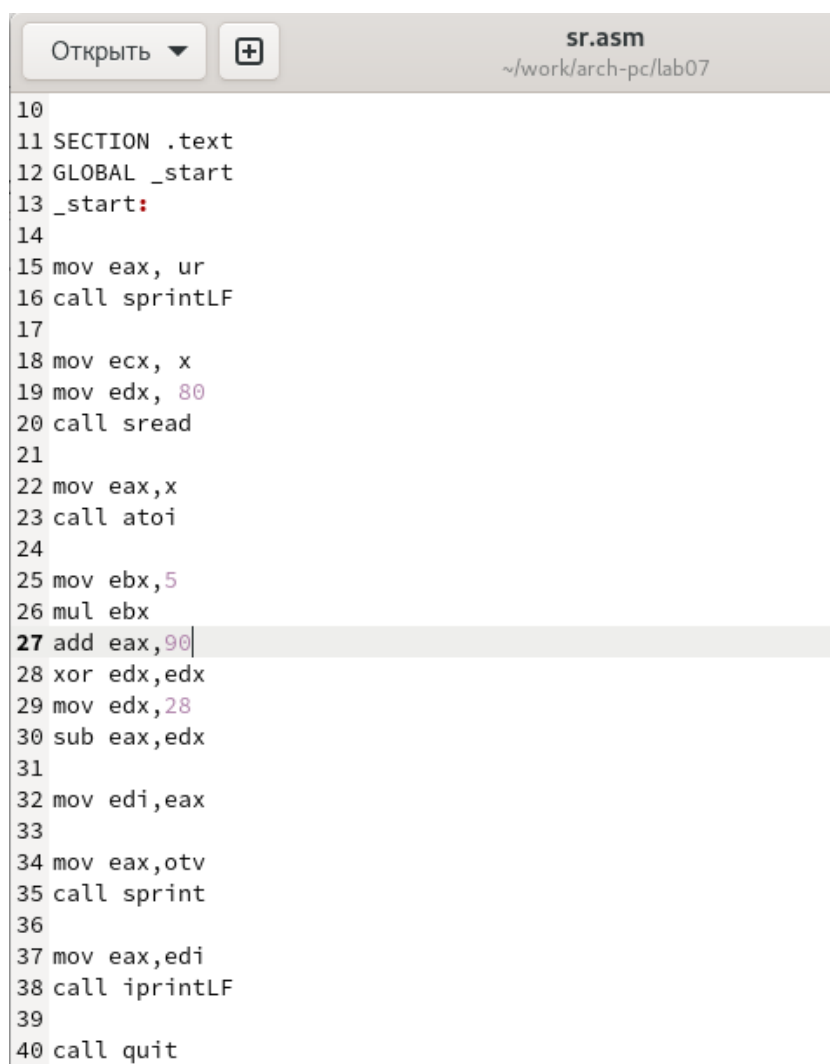
1. Перед началом работы создала файл. Написала программу для вычисления выражения из варианта 10, который был получен ранее при выполнении лабораторной работы.





```
1 %include 'in_out.asm'
2
3 SECTION .data
4 ur: DB '5(x+18)-28 = (5x+90)-28',0
5 per: DB 'Введите значение x: ',0
6 otv: DB 'Ответ: ',0
7
8 SECTION .bss
9 x: RESB 80
10
11 SECTION .text
12 GLOBAL _start
13 _start:
14
15 mov eax, ur
16 call sprintLF
17
18 mov ecx, x
19 mov edx, 80
20 call sread
21
22 mov eax,x
23 call atoi
24
25 mov ebx,5
26 mul ebx
27 add eax,90
28 xor edx,edx
29 mov edx,28
30 sub eax,edx
31
```

Рис. 5.1: Программа



```
10
11 SECTION .text
12 GLOBAL _start
13 _start:
14
15 mov eax, ur
16 call sprintf
17
18 mov ecx, x
19 mov edx, 80
20 call sread
21
22 mov eax, x
23 call atoi
24
25 mov ebx, 5
26 mul ebx
27 add eax, 90
28 xor edx, edx
29 mov edx, 28
30 sub eax, edx
31
32 mov edi, eax
33
34 mov eax, otv
35 call sprint
36
37 mov eax, edi
38 call iprintLF
39
40 call quit
```

Рис. 5.2: Программа

```
[emmednikova@fedora lab07]$ gedit sr.asm
[emmednikova@fedora lab07]$ nasm -f elf sr.asm
[emmednikova@fedora lab07]$ ld -m elf_i386 -o sr sr.o
[emmednikova@fedora lab07]$ ./sr
5(x+18)-28 = (5x+90)-28
2
Ответ: 72
[emmednikova@fedora lab07]$ ./sr
5(x+18)-28 = (5x+90)-28
3
Ответ: 77
[emmednikova@fedora lab07]$
```

Рис. 5.3: Результат

## 6 Выводы

Освоила арифметические инструкции языка ассемблера NASM.