

# **Отчёта по лабораторной работе №6**

**Арифметические операции в NASM**

Хохлачева Полина Дмитриевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Ответы на вопросы по программе</b>	<b>13</b>
<b>4</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

# Список иллюстраций

2.1	Создаём с помощью команды <code>mkdir</code> . . . . .	6
2.2	Заполняем файл . . . . .	6
2.3	Запускаем файл и смотрим работу . . . . .	7
2.4	Убираем кавычки . . . . .	7
2.5	Запускаем смотрим работу . . . . .	7
2.6	Создаём файл . . . . .	7
2.7	Заполняем файл . . . . .	8
2.8	Смотрим на работу команды . . . . .	8
2.9	Убираем кавычки . . . . .	8
2.10	Смотрим на работу программы . . . . .	9
2.11	Изменяем файл . . . . .	9
2.12	Смотрим на работу программы . . . . .	9
2.13	Создаём файл . . . . .	9
2.14	Заполняем файл . . . . .	10
2.15	Смотрим на результат работы программы . . . . .	10
2.16	Редактируем файл . . . . .	11
2.17	Смотрим на результат работы . . . . .	11
2.18	Создаём файл . . . . .	11
2.19	Смотрим на результат работы . . . . .	12
2.20	Проверяем работу программы . . . . .	12
3.1	Создаём файл . . . . .	13
3.2	Создаём файл . . . . .	14
3.3	Создаём файл . . . . .	14
3.4	Создаём файл . . . . .	14

## **Список таблиц**

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Выполнение лабораторной работы

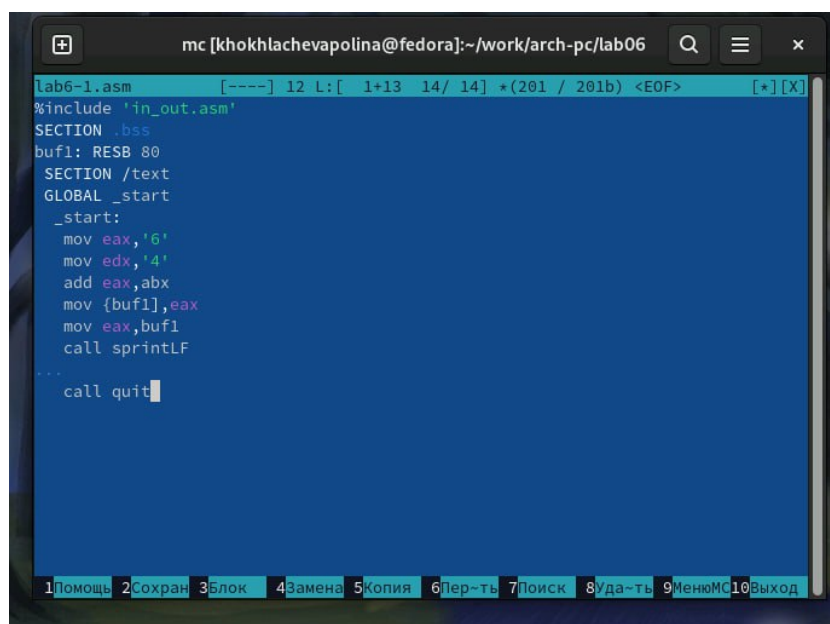
Создаём каталог для программ лабораторной работы №6



```
khokhlachevapolina@fedora:~/work/arch-pc/lab06
khokhlachevapolina@fedora:~$ mkdir ~/work/arch-pc/lab06
khokhlachevapolina@fedora:~$ cd ~/work/arch-pc/lab06
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ touch lab6-1.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$
```

Рис. 2.1: Создаём с помощью команды mkdir

Открываем файл и заполняем в соответствии с листингом



```
lab6-1.asm  [-----] 12 L: [ 1+13 14/ 14] *(201 / 201b) <EOF> [*][X]
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION /text
GLOBAL _start
_start:
    mov eax, '6'
    mov edx, '4'
    add eax, ebx
    mov [buf1], eax
    mov eax, buf1
    call sprintLF
    ...
    call quit
```

Рис. 2.2: Заполняем файл

Создаём исполняемый файл и запускаем его

```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ mc
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ mc
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./lab6-1
6
khokhlachevapolina@fedora:~/work/arch-pc/lab06$
```

Рис. 2.3: Запускаем файл и смотрим работу

Открываем файл для редактирования

```
lab6-1.asm [-M--] 12 L:[ 1+ 9 10/ 16] *(123 / 202b) 0010 0x00A
#include 'in_out.asm'

SECTION .bss
buf1: > RESB 80
SECTION .text
GLOBAL _start
_start:
...
mov eax,6
mov edx,4
add edx,edx
mov [buf1],eax
mov eax,buf1
call sprintLF
...
call quit
```

Рис. 2.4: Убираем кавычки

Создаём файл и запускаем его

```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./lab6-1
khokhlachevapolina@fedora:~/work/arch-pc/lab06$
```

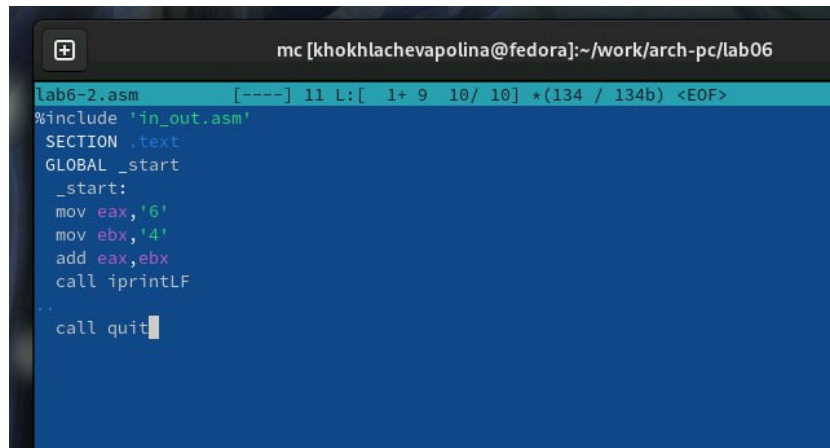
Рис. 2.5: Запускаем смотрим работу

Создаём новый файл в каталоге

```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ mc
```

Рис. 2.6: Создаём файл

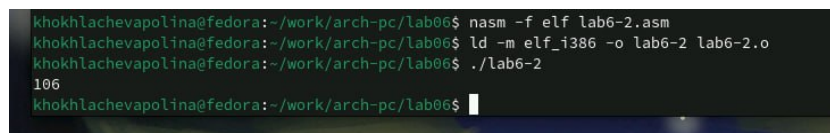
Заполняем файл в соответствие с листингом

A screenshot of a code editor window titled 'mc [khokhlachevapolina@fedora]:~/work/arch-pc/lab06'. The editor shows the file 'lab6-2.asm' with a status bar indicating line 11, column 10. The code is as follows:

```
lab6-2.asm [----] 11 L: [ 1+ 9 10/ 10] *(134 / 134b) <EOF>
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 2.7: Заполняем файл

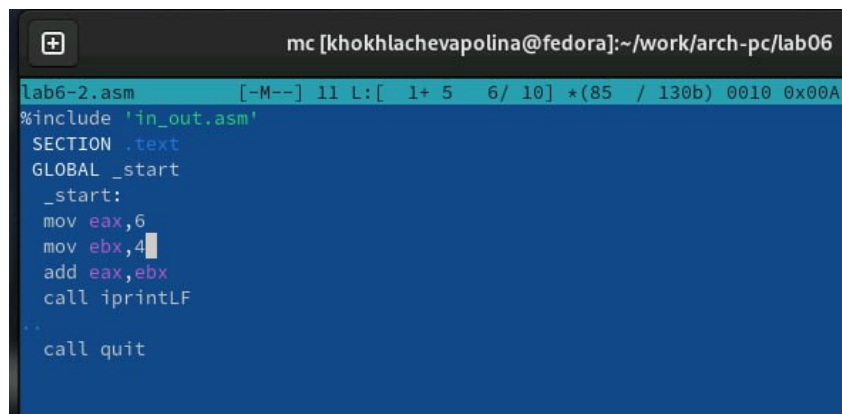
Создаём исполняемый файл и запускаем его

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

```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./lab6-2
106
khokhlachevapolina@fedora:~/work/arch-pc/lab06$
```

Рис. 2.8: Смотрим на работу команды

Открываем файл и редактируем его

A screenshot of a code editor window titled 'mc [khokhlachevapolina@fedora]:~/work/arch-pc/lab06'. The editor shows the file 'lab6-2.asm' with a status bar indicating line 11, column 5. The code is as follows:

```
lab6-2.asm [-M--] 11 L: [ 1+ 5 6/ 10] *(85 / 130b) 0010 0x00A
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 2.9: Убираем кавычки

Создаём исполняемый файл и запускаем его



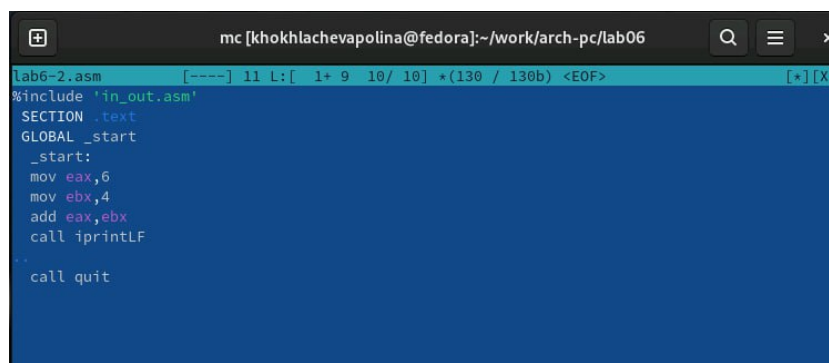
```

hokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
hokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
hokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./lab6-2
0
hokhlachevapolina@fedora:~/work/arch-pc/lab06$

```

Рис. 2.10: Смотрим на работу программы

Открываем файл и редактируем его



```

lab6-2.asm [----] 11 L: [ 1+ 9 10/ 10] *(130 / 130b) <EOF> [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
..
call quit

```

Рис. 2.11: Изменяем файл

Создаём исполняемый файл и запускаем его

```

hokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
hokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
hokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./lab6-2
0
hokhlachevapolina@fedora:~/work/arch-pc/lab06$

```

Рис. 2.12: Смотрим на работу программы

#Выполнение арифметических операций NASM

Создаём новый файл в каталоге

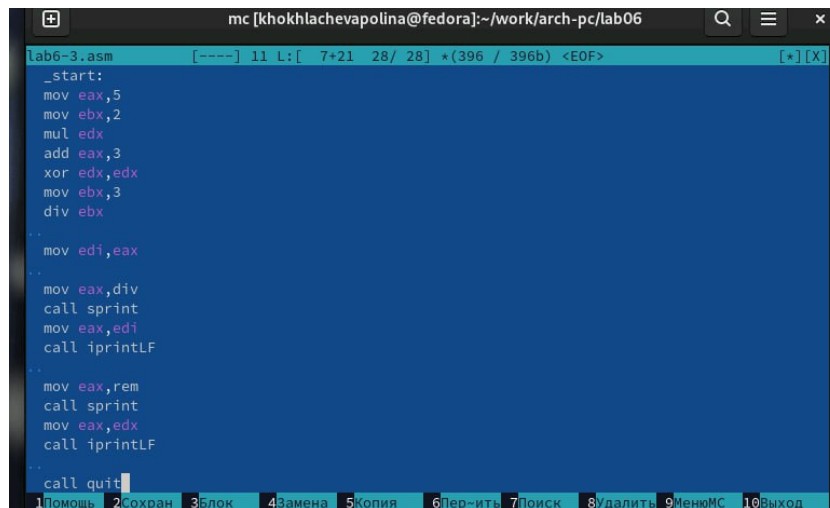
```

khokhlachevapolina@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$

```

Рис. 2.13: Создаём файл

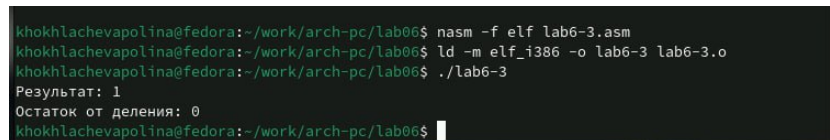
Открываем файл и редактируем в соответствии с листингом



```
lab6-3.asm [----] 11 L: [ 7+21 28/ 28] *(396 / 396b) <EOF> [*][X]
_start:
mov eax,5
mov ebx,2
mul edx
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. 2.14: Заполняем файл

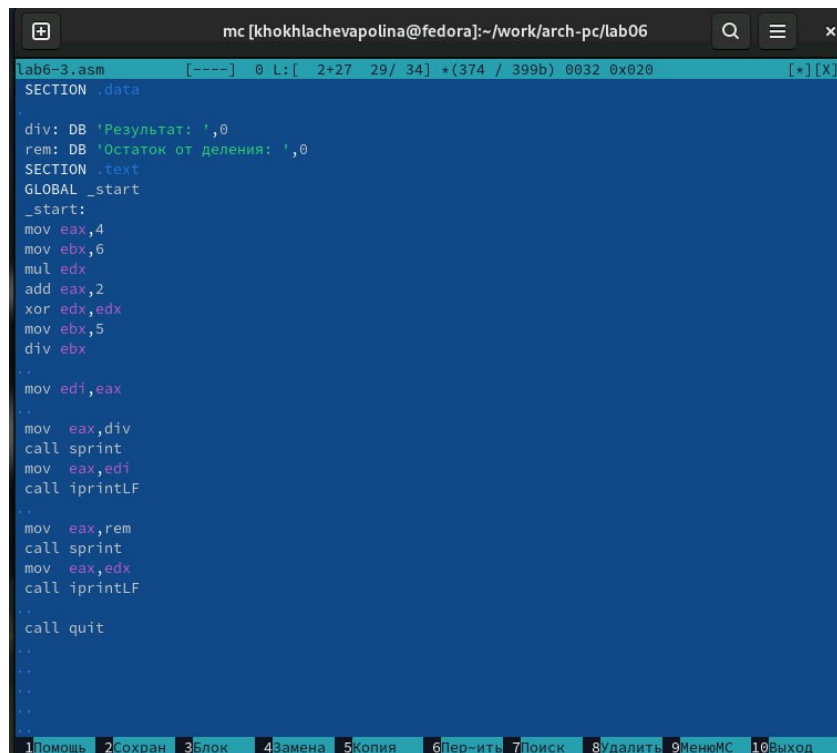
Создаём исполняемый файл и запускаем его



```
khokhlachevapolina@fedora: ~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
khokhlachevapolina@fedora: ~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
khokhlachevapolina@fedora: ~/work/arch-pc/lab06$ ./lab6-3
Результат: 1
Остаток от деления: 0
khokhlachevapolina@fedora: ~/work/arch-pc/lab06$
```

Рис. 2.15: Смотрим на результат работы программы

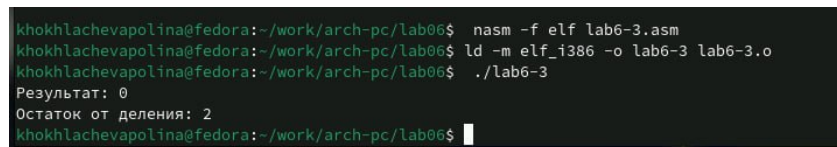
Создаём файл для вычисления выражения  $\square(\square) = (4 \square 6 + 2)/5$ .



```
lab6-3.asm [----] 0 L: [ 2+27 29/ 34] *(374 / 399b) 0032 0x020 [*] [X]
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,6
mul edx
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. 2.16: Редактируем файл

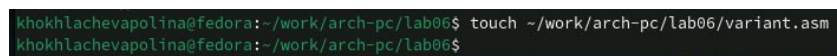
Компилируем файл и запускаем программу



```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 0
Остаток от деления: 2
khokhlachevapolina@fedora:~/work/arch-pc/lab06$
```

Рис. 2.17: Смотрим на результат работы

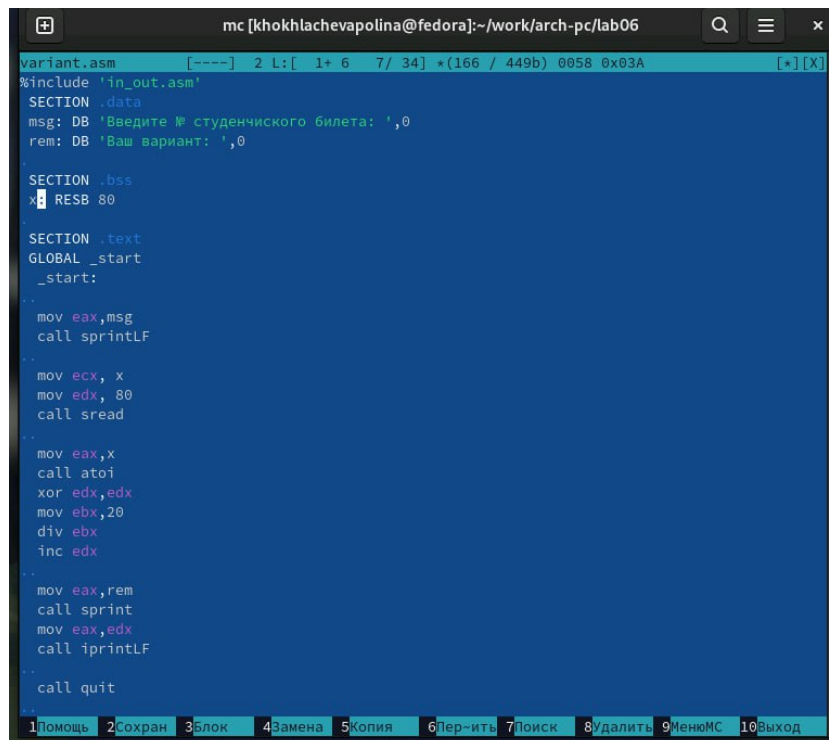
Создаём новый файл в каталоге



```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$
```

Рис. 2.18: Создаём файл

Открываем файл и редактируем в соответствии с листингом



```
variant.asm [----] 2 L: [ 1+ 6 7/ 34] *(166 / 449b) 0058 0x03A [*] [X]
#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 sprintf

mov ecx,x
mov edx,80
call sread

mov eax,x
call atoi
xor edx,edx
mov ebx,20
div ebx
inc edx

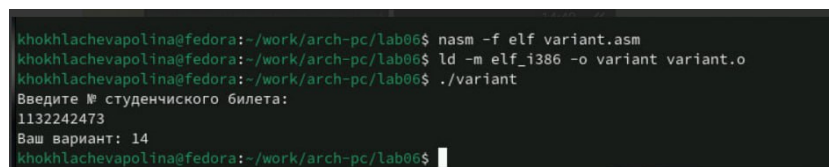
mov eax,rem
call sprintf
mov eax,edx
call iprintf

call quit
```

1 Помощь 2 Сохран 3 Блок 4 Замена 5 Копия 6 Пер-ить 7 Поиск 8 Удалить 9 МенюМС 10 Выход

Рис. 2.19: Смотрим на результат работы

Компилируем файл и запускаем его



```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132242473
Ваш вариант: 14
khokhlachevapolina@fedora:~/work/arch-pc/lab06$
```

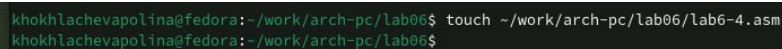
Рис. 2.20: Проверяем работу программы

### 3 Ответы на вопросы по программе

1. Строка 'mov eax,msg' и 'call sprintLF'
2. Эти инструкции используются для считывания строки, введённой пользователем, и её сохранения в памяти по адресу x. • mov ecx, x — задаёт адрес для сохранения строки. • mov edx, 80 — задаёт максимальную длину считываемой строки. • call sread — вызывает процедуру чтения строки из стандартного ввода.
3. Инструкция call atoi используется для преобразования строки (ASCII-кодов), введённой пользователем, в число, которое будет сохранено в регистре eax.
4. xor edx, edx mov ebx, 20 div ebx inc edx
4. Остаток от деления записывается в регистр edx
5. Инструкция inc edx увеличивает значение в регистре edx на 1. В данном случае это используется для прибавления 1 к остатку от деления, чтобы получить номер варианта.
6. mov eax, rem call sprint mov eax, edx call iprintLF

#Задание для самостоятельной работы

Создаём новый файл в каталоге



```
khokhlachevapolina@fedora: ~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-4.asm
khokhlachevapolina@fedora: ~/work/arch-pc/lab06$
```

Рис. 3.1: Создаём файл

Открываем его и заполняем, чтобы решалось выражение  $(x/2+8)*3$

```
lab6-4.asm [----] 2 L: [ 7+31 38/ 39] *(506 / 518b) 0010 0x00A [*][X]
x: RESB 80
res: 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
mov ebx, eax

xor edx, edx
mov eax, ebx
mov ecx, 2
div ecx
add eax, 8
imul eax, 3
sub eax, 14

mov ebx, eax
mov eax, result
call sprint
mov eax, ebx
call iprintfLF

call quit
```

Рис. 3.2: Создаём файл

Компилируем программу и проверяем для x=1

```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
1
Результат вычислений: 10
```

Рис. 3.3: Создаём файл

Компилируем программу и проверяем для x=4

```
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
khokhlachevapolina@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите x:
4
Результат вычислений: 16
khokhlachevapolina@fedora:~/work/arch-pc/lab06$
```

Рис. 3.4: Создаём файл

## 4 Выводы

Мы приобрели навыки создания исполнительный файлов для решения выражений и освоили арифметический инструкции

## **Список литературы**