

# **Отчет по лабораторной 6**

**Архитектура вычислительных систем**

Ким Эрика Алексеевна

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	19

## Список иллюстраций

3.1	Текст . . . . .	8
3.2	проверка . . . . .	8
3.3	исправляем . . . . .	9
3.4	проверка . . . . .	9
3.5	файл . . . . .	10
3.6	тс . . . . .	10
3.7	текст . . . . .	11
3.8	проверка . . . . .	11
3.9	исправляем . . . . .	12
3.10	проверка . . . . .	12
3.11	исправляем . . . . .	12
3.12	каталог . . . . .	13
3.13	проверка . . . . .	13
3.14	текст . . . . .	14
3.15	исправляем . . . . .	14
3.16	исправляем . . . . .	15
3.17	команда . . . . .	15
3.18	файл . . . . .	15
3.19	проверка . . . . .	16
3.20	текст . . . . .	16
3.21	ввод данных . . . . .	17

## Список таблиц

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

## 2 Задание

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 6.3 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 6.3

### 3 Выполнение лабораторной работы

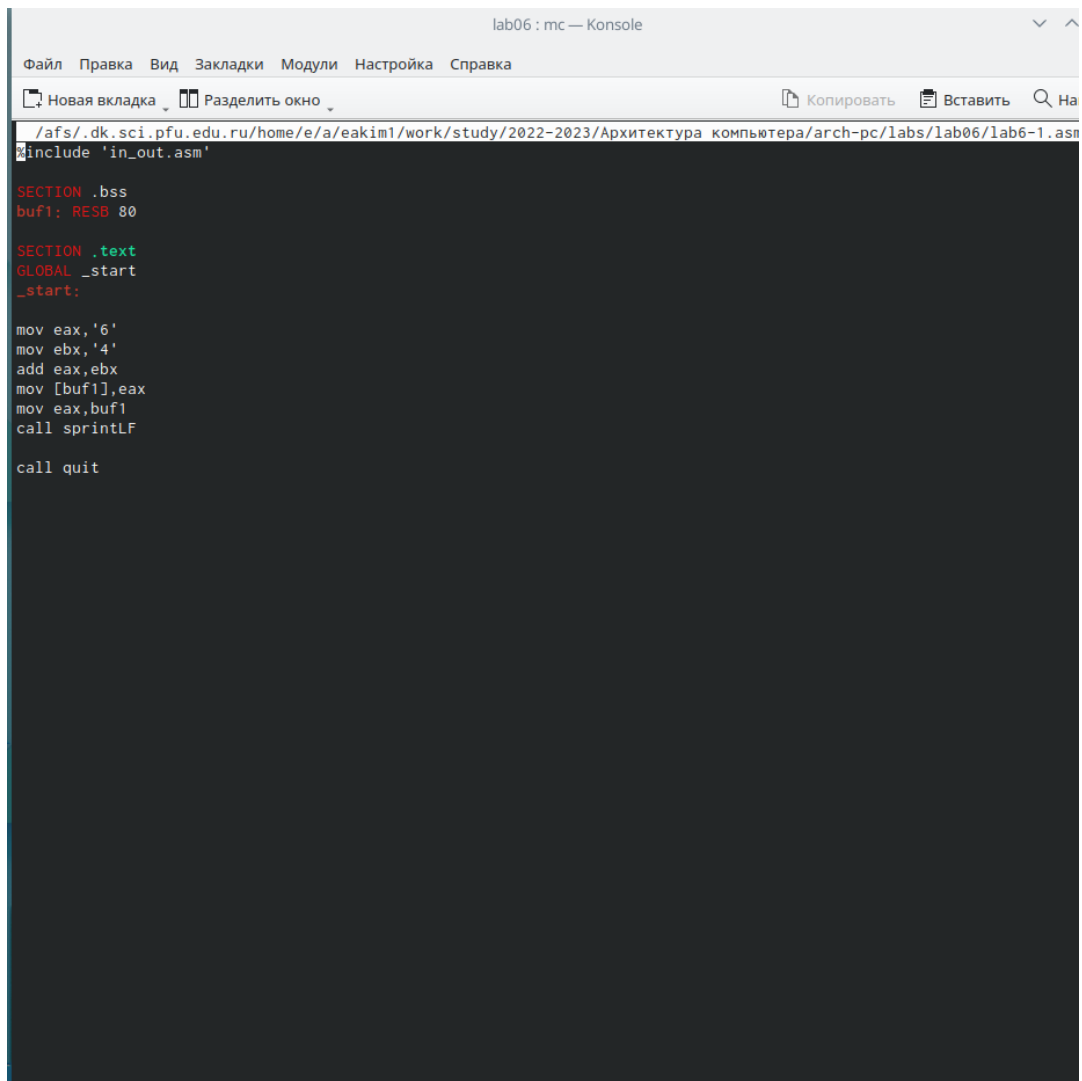
1. Создаем каталог, переходим в него и создаем файл lab6-1.asm

[каталог] (image/рис1.png){ #fig:001 width=90% }

2. Открываем mc и проверяем

mc

3. Вставляем заданные текст



```
lab06 : mc — Konsole

Файл  Правка  Вид  Закладки  Модули  Настройка  Справка

Новая вкладка  Разделить окно  Копировать  Вставить  На

/afs/.dk.sci.pfu.edu.ru/home/e/a/eakim1/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06/lab6-1.asm
#include 'in_out.asm'

SECTION .bss
buf1: RESB 80

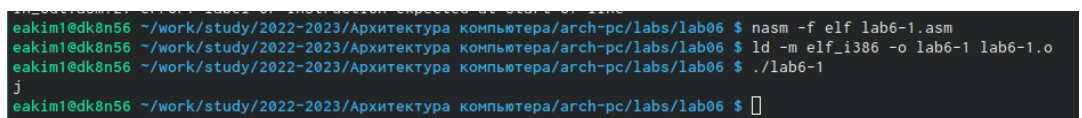
SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 3.1: Текст

#### 4. Проверяем работает ли команда



```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-1
j
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.2: проверка

#### 5. Исправляем текст



```
lab06: mc — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Поиск
...k.sci.pfu.edu.ru/home/e/a/eakim1/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06/lab6-1.asm Изменен
#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 sprintf
call quit
```

Рис. 3.3: исправляем

## 6. Проверяем

```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-1

eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.4: проверка

## 7. Создаем файл lab6-2.asm

```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ touch lab6-2.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc
```

Рис. 3.5: файл

## 8. Открываем mc и проверяем создался ли файл

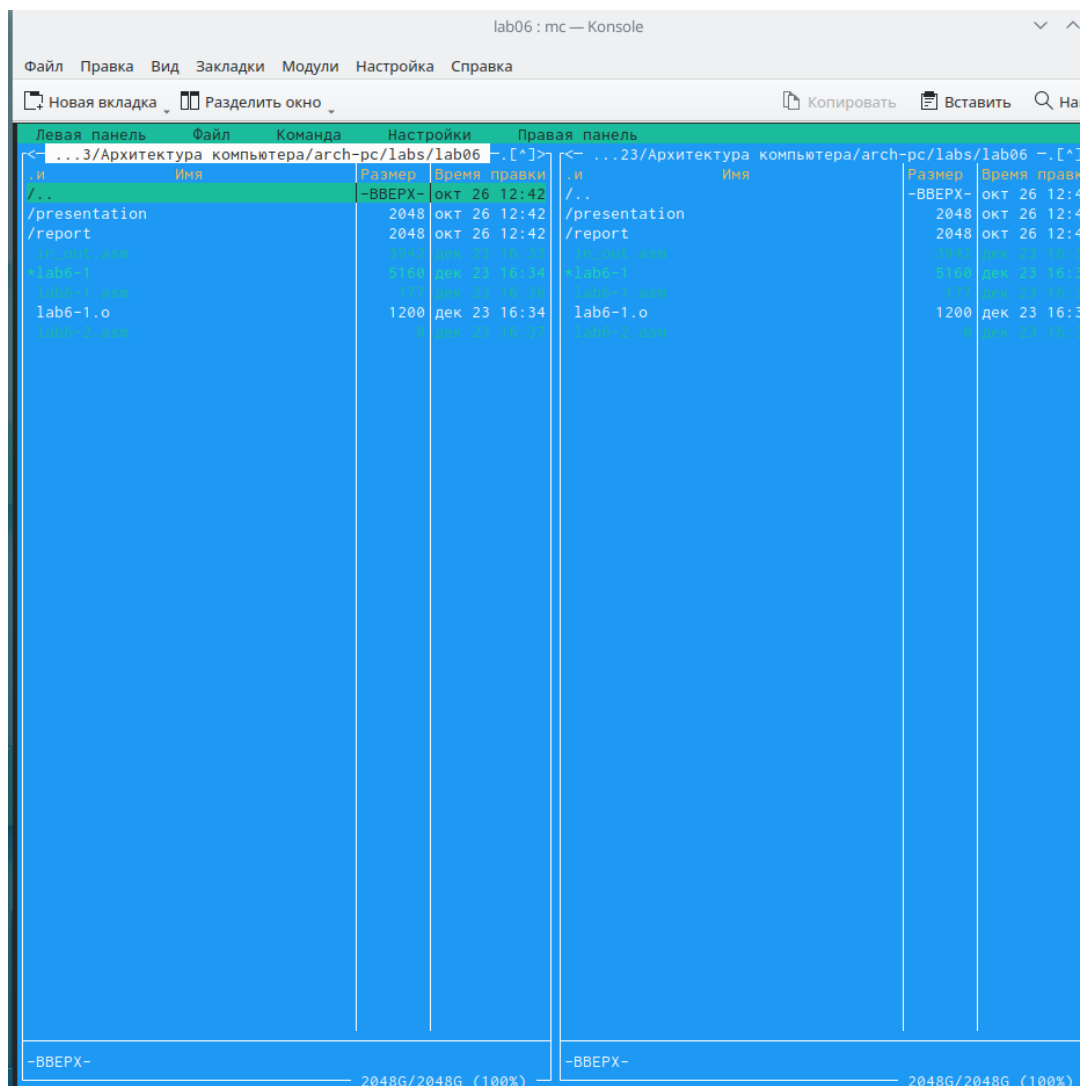


Рис. 3.6: mc

## 9. Вставляем текст

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

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

call quit
```

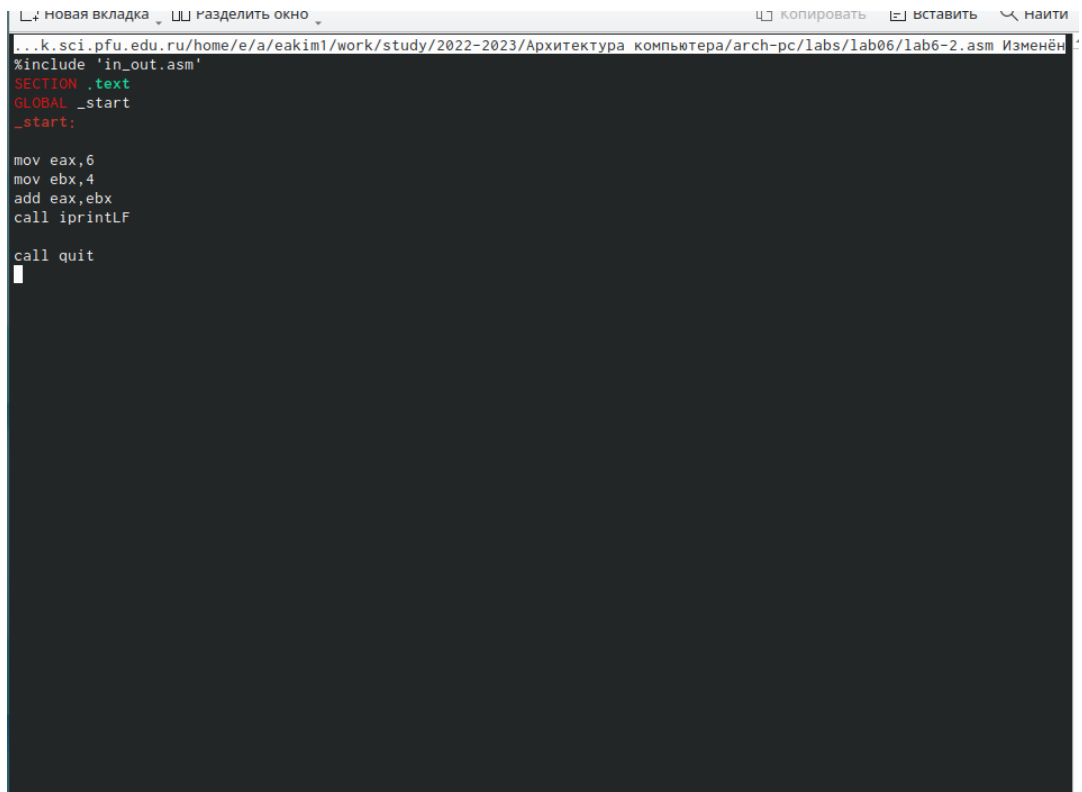
Рис. 3.7: текст

## 10. Проверяем работает ли команда

```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
106
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.8: проверка

## 11. Исправляем текст во втором файле



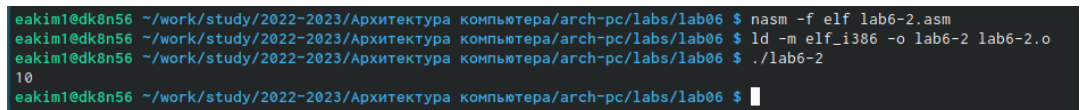
```
..k.sci.pfu.edu.ru/home/e/a/eakim1/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06/lab6-2.asm Изменён
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

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

call quit
```

Рис. 3.9: исправляем

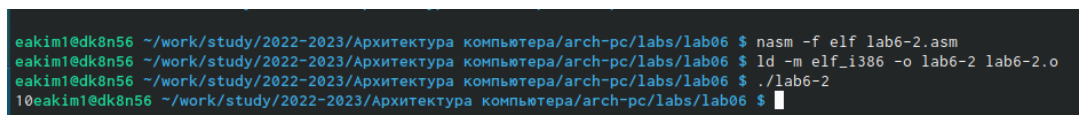
## 12. Проверяем команду



```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
10
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.10: проверка

## 13. Исправляем iprintLF на iprint



```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
10
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.11: исправляем

## 14. Создаем каталог lab6-3.asm

```
10eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ touch lab6-3.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc
```

Рис. 3.12: каталог

## 15. Проверяем созданся ли каталог в mc

Левая панель				Правая панель			
Файл	Команда	Настройки		Файл	Команда	Настройки	
..	Имя	Размер	Время правки	..	Имя	Размер	Время правки
..	-ВВЕРХ-	офт 26 12:42		..	-ВВЕРХ-	офт 26 12:42	
/presentation		2048	офт 26 12:42	/presentation		2048	офт 26 12:42
/report		2048	дек 23 17:55	/report		2048	дек 23 17:55
in_out.asm		3940	дек 23 16:33	in_out.asm		3940	дек 23 16:33
*lab6-1		5160	дек 23 16:31	*lab6-1		5160	дек 23 16:31
lab6-1.asm		1773	дек 23 16:38	lab6-1.asm		1773	дек 23 16:38
lab6-1.o		1200	дек 23 16:41	lab6-1.o		1200	дек 23 16:41
*lab6-2		5060	дек 23 16:45	*lab6-2		5060	дек 23 16:45
lab6-2.asm		1114	дек 23 16:45	lab6-2.asm		1114	дек 23 16:45
lab6-2.o		1040	дек 23 16:45	lab6-2.o		1040	дек 23 16:45
*lab6-3		4072	дек 23 17:03	*lab6-3		4072	дек 23 17:03
lab6-3.asm		1405	дек 23 17:03	lab6-3.asm		1405	дек 23 17:03
lab6-3.o		1328	дек 23 17:03	lab6-3.o		1328	дек 23 17:03

Рис. 3.13: проверка

## 16. Вставляем текст

```

;-----
; Программа вычисления выражения
;-----

%include 'in_out.asm' ; подключение внешнего файла

SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления

mov edi,eax ; запись результата вычисления в 'edi'

; ---- Вывод результата на экран

mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов

mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов

call quit ; вызов подпрограммы завершения

```

Рис. 3.14: текст

## 17. Проверяем команду

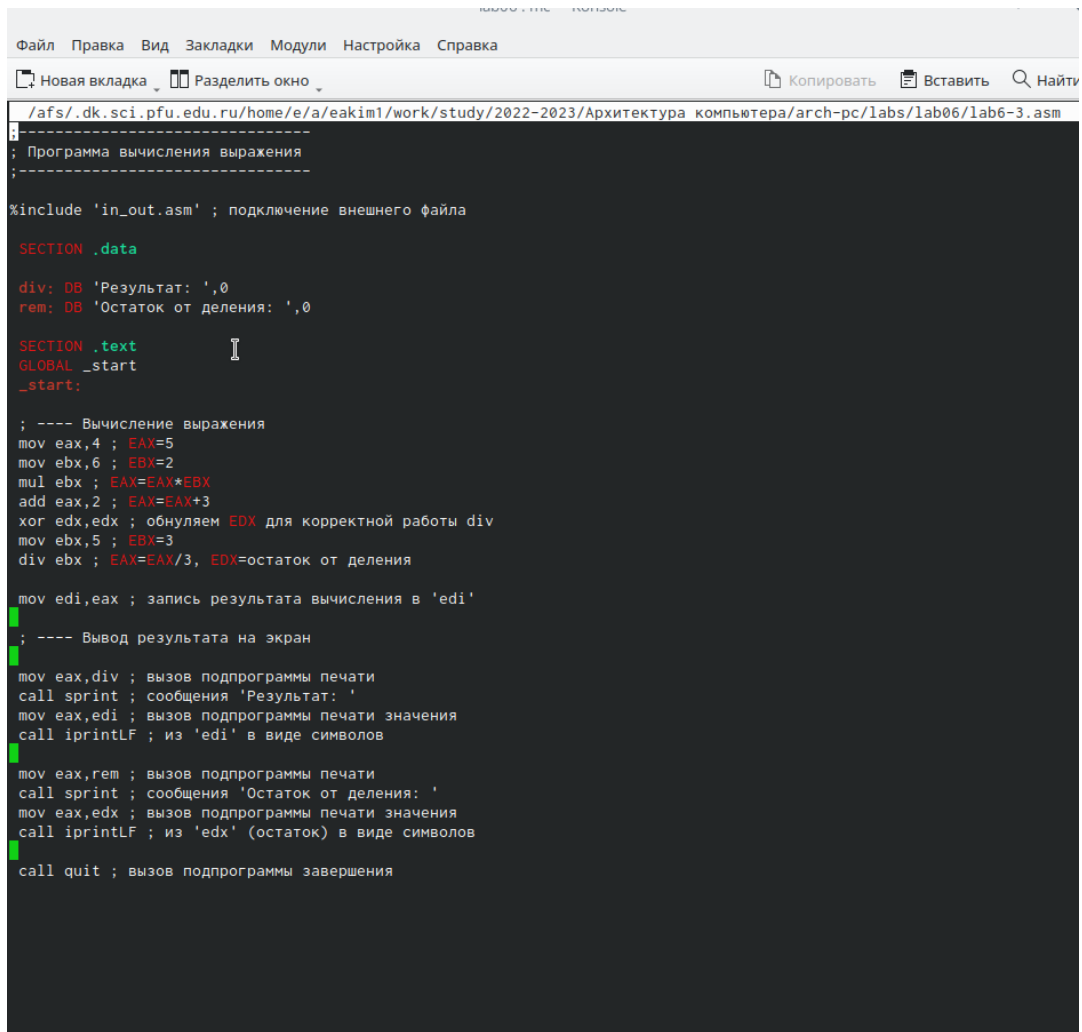
```

eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-3.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $

```

Рис. 3.15: исправляем

## 18. Исправляем текст



```
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
/afs/.dk.sci.pfu.edu.ru/home/e/a/eakim1/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06/lab6-3.asm
; Программа вычисления выражения
; -----
%include 'in_out.asm' ; подключение внешнего файла

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

SECTION .text
GLOBAL _start
_start:

; ---- Вычисление выражения
mov eax,4 ; EAX=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления

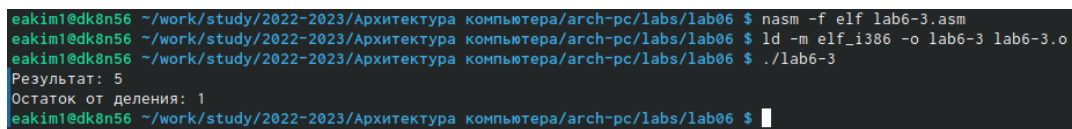
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintf ; из 'edi' в виде символов

mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintf ; из 'edx' (остаток) в виде символов

call quit ; вызов подпрограммы завершения
```

Рис. 3.16: исправляем

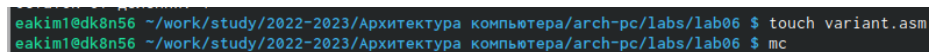
## 19. Выводим команду



```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-3.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.17: команда

## 20. Создаем файл variant



```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ touch variant.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ls
```

Рис. 3.18: файл

## 21. Проверяем в тс создан ли файл

variant	8100	дек 23 16:56	variant	8100	дек 23 16:56
variant.asm	810	дек 23 16:56	variant.asm	810	дек 23 16:56
variant.o	1440	дек 23 16:56	variant.o	1440	дек 23 16:56

Рис. 3.19: проверка

## 22. Вводим текст

```
;-----  
; Программа вычисления варианта  
;-----  
%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 ; 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 sprintf  
mov eax, edx  
call iprintf  
call quit
```

Рис. 3.20: текст



## 23. Проверяем команду и вводим данные

```
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf variant.asm
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o variant variant.o
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./variant
Введите No студенческого билета:
1132229054
Ваш вариант: 15
eakim1@dk8n56 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.21: ввод данных

### Вопросы

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

Ответ: `mov eax,rem call sprint`

2. Для чего используются следующие инструкции? `nasm mov ecx, x mov edx, 80 call sread`

Ответ: `nasm` - переход к языку ассемблера `mov ecx, x` - присвоение значения `x` переменной `ecx` `mov edx, 80` - присвоение значение 80 переменной `edx` `call sread` - для считывания в переменную какого то числа

3. Для чего используется инструкция "call atoi"? Ответ: Конвертирует строку, на которую указывает параметр `str`, в величину типа `int`
4. Какие строки листинга 7.4 отвечают за вычисления варианта?

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

5. В какой регистр записывается остаток от деления при выполнении инструкции "div ebx"?

Ответ: Остаток запишется в регистр `dx`

6. Для чего используется инструкция "inc edx"?

Ответ: Это инкремент для прибавления единицы к переменной

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

Ответ: `mov eax,rem call sprint mov eax,edx call iprintLF`

## 4 Выводы

Освоили арифметические инструкции языка ассемблера NASM.