

лабораторная работа №6.

Арифметические операции в NASM.

Митрофанов Тимур Александрович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Ответ на вопросы	14
3	Задание для самостоятельной работы	16
4	Выводы	19

Список иллюстраций

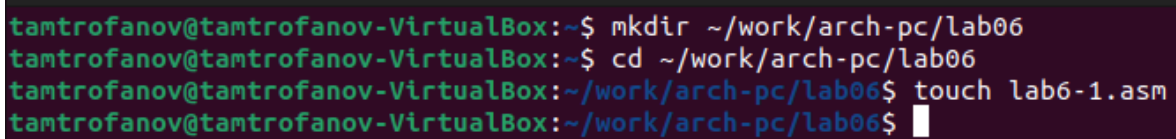
2.1	Создание подкаталога и файла в нём	5
2.2	Содержимое файла <i>lab6-1.asm</i>	6
2.3	Компиляция файла <i>lab6-1.asm</i> и запуск кода	6
2.4	Внесённые изменения в файл <i>lab6-1.asm</i>	7
2.5	Компиляция изменённого файла <i>lab6-1.asm</i> и запуск кода	7
2.6	Вырезка из таблицы ASCII	7
2.7	Создание файла <i>lab6-2.asm</i>	8
2.8	Внесённый код в файл <i>lab6-2.asm</i>	8
2.9	Компиляция файла <i>lab6-2.asm</i> и запуск кода	8
2.10	Внесённые изменения в файл <i>lab6-3.asm</i>	9
2.11	Компиляция файла изменённого <i>lab6-3.asm</i> и запуск кода	9
2.12	Изменение кода файла <i>lab6-3.asm</i>	10
2.13	Компиляция изменённого файла <i>lab6-3.asm</i> и запуск кода	10
2.14	Создание файла файл <i>lab6-3.asm</i>	10
2.15	Внесённый код в файл <i>lab6-3.asm</i>	11
2.16	Компиляция файла <i>lab6-3.asm</i> и запуск кода	11
2.17	Изменённый файл <i>lab6-3.asm</i>	12
2.18	Компиляция изменённого файла <i>lab6-3.asm</i> и запуск кода	12
2.19	Создание файла файл <i>variant.asm</i>	13
2.20	Внесённый код в файл <i>variant.asm</i>	13
2.21	Компиляция файла <i>variant.asm</i> и запуск кода	14
2.22	Аналитическая проверка работы кода	14
3.1	Создание файла файл <i>sam_rabota.asm</i>	16
3.2	Внесённый код в файл <i>sam_rabota.asm</i>	17
3.3	Компиляция файла <i>sam_rabota.asm</i> , запуск кода и проверка кода	18

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

При помощи стандартных команд **mkdir**, **cd** и **touch** создал подкаталог `~/work/arch-pc/lab06`, а в нём файл `lab6-1.asm` (рис. 2.1).



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

Рис. 2.1: Создание подкаталога и файла в нём

В созданный файл `lab6-1.asm` скопировал код программы из листинга 6.1 (рис. 2.2). Затем скомпелировал этот файл и запустил для проверки его работы (рис. 2.3). В результате выполнения кода получил ответ - *j*.

```
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 sprintf
13 call quit
```

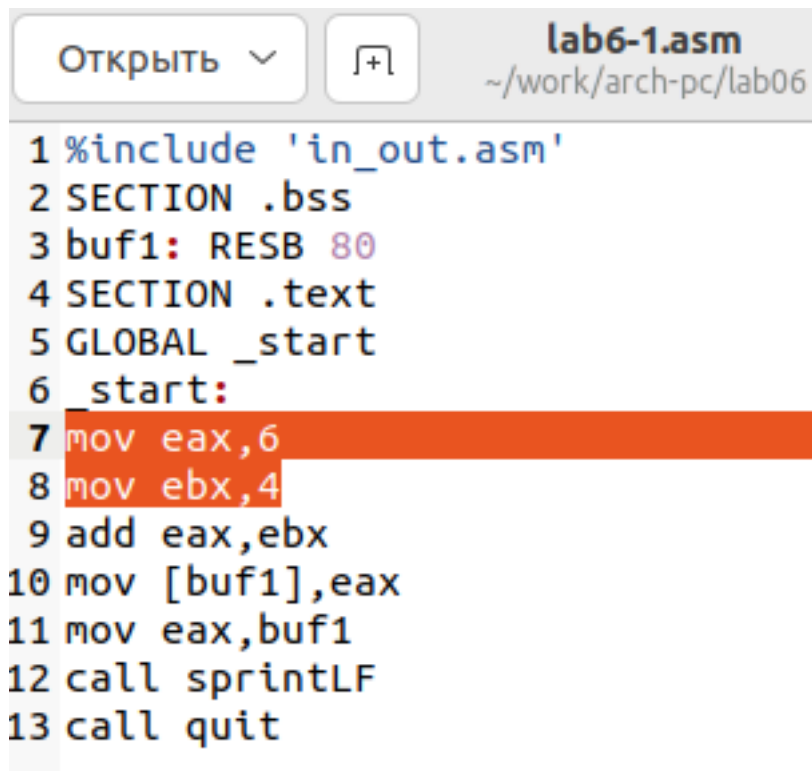
Рис. 2.2: Содержимое файла *lab6-1.asm*

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

Рис. 2.3: Компиляция файла *lab6-1.asm* и запуск кода

Заменяю несколько строчек кода в файле *lab6-1.asm* (рис. 2.4). Потом скомпилировал вновь изменённый файл и проверил его работу (рис. 2.5).

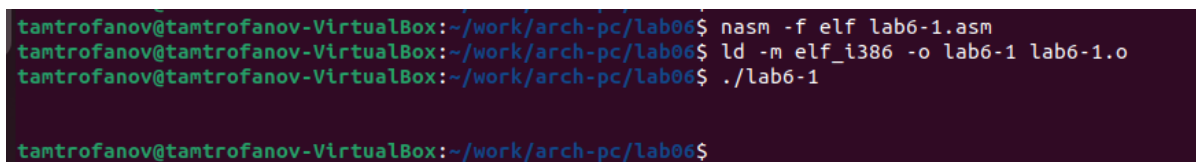
Код *10* означает пререход на новую строку (рис. 2.6). Этот символ не отображался в привычном понимании как тест, но исполнился, переведя вывод консоли на новую строку.



```
lab6-1.asm
~/work/arch-pc/lab06

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 sprintf
13 call quit
```

Рис. 2.4: Внесённые изменения в файл *lab6-1.asm*



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

Рис. 2.5: Компиляция изменённого файла *lab6-1.asm* и запуск кода

9	11	0x09	1001	HT, \t
10	12	0x0A	1010	LF, \n
11	13	0x0B	1011	VT

Рис. 2.6: Вырезка из таблицы **ASCII**

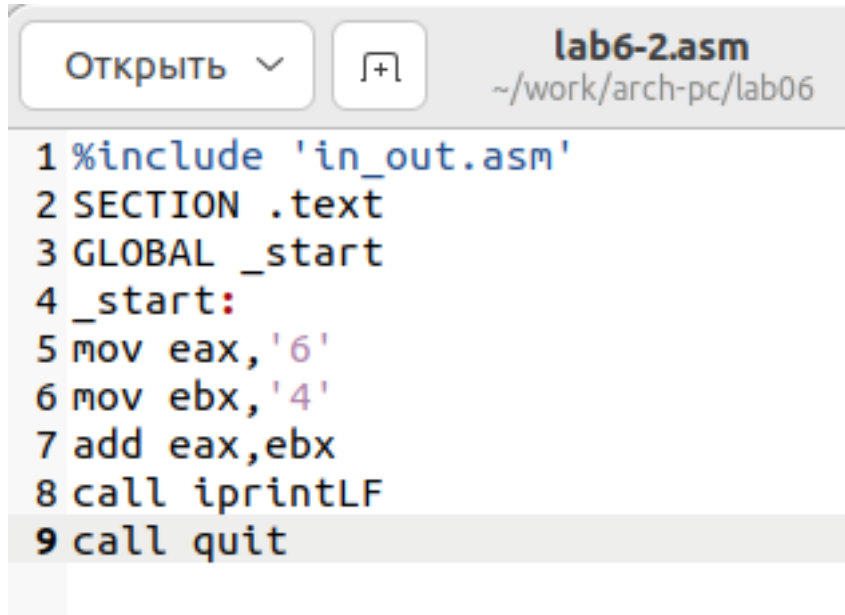
Теперь создал файл *lab6-2.asm* (рис. 2.7). Добавил в него код из листинга 6.2 (рис. 2.8). Скомпелировал его и запустил для проверки работы (рис. 2.9).

```

-VirtualBox:~/work/arch-pc/lab06$
-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 2.7: Создание файла *lab6-2.asm*



```

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. 2.8: Внесённые код в файл *lab6-2.asm*

```

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

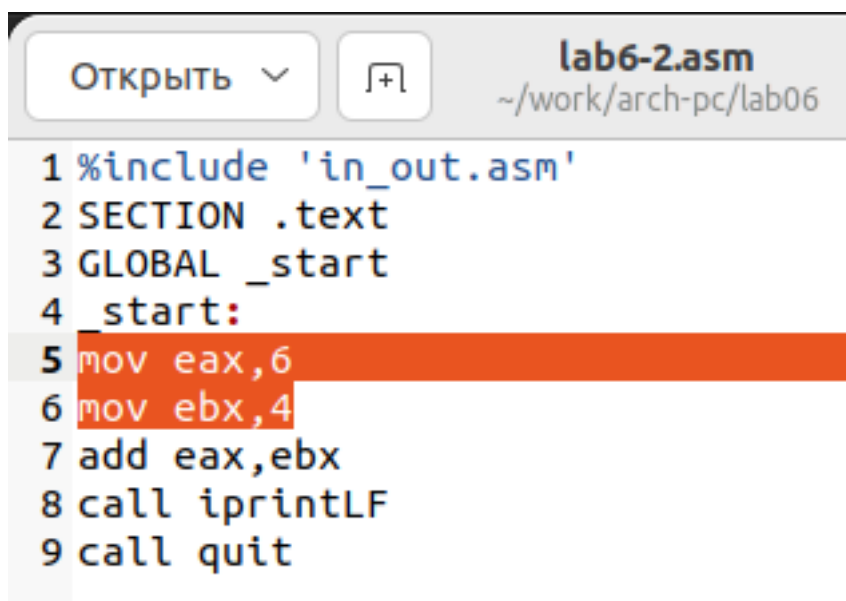
```

Рис. 2.9: Компиляция файла *lab6-2.asm* и запуск кода

Согласно инструкции изменил часть кода в файле *lab6-2.asm* (рис. 2.10). Потом скопировал его и запустил код для проверки (рис. 2.11). При исполнении программы был получен результат - 10.

Заменяю функцию **iprintLF** на **iprint** (рис. 2.12). Компилирую файл и запускаю код для проверки (рис. 2.13). На этот раз после вывода результат консоль не

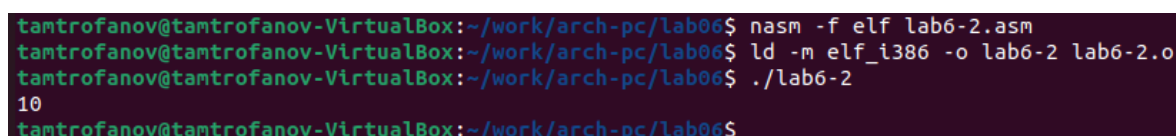
стала переходить на новую строку ведь функция **iprint** не персматривает этого, в отличии от функции **iprintLF**.



```
lab6-2.asm
~/work/arch-pc/lab06

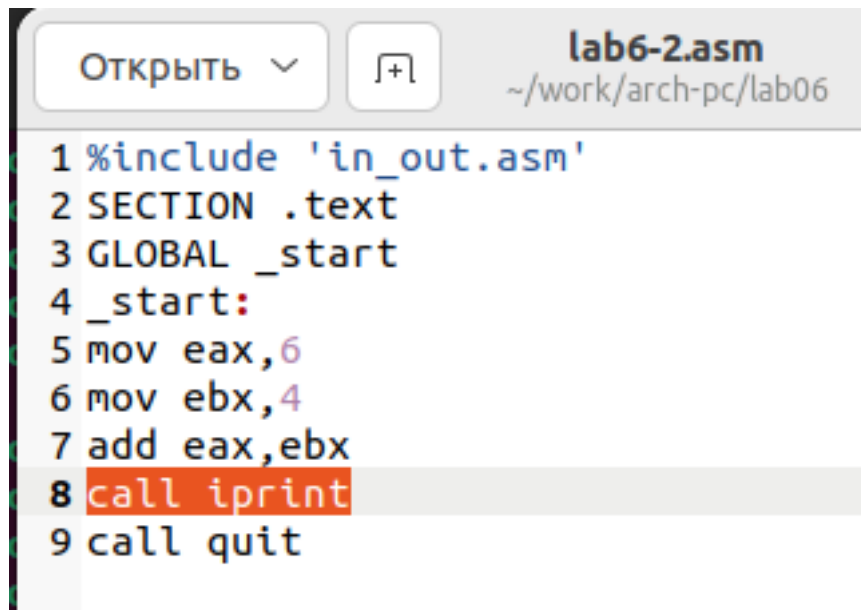
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit
```

Рис. 2.10: Внесённые изменения в файл *lab6-3.asm*



```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.11: Компиляция файла изменённого *lab6-3.asm* и запуск кода



```
lab6-2.asm
~/work/arch-pc/lab06

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. 2.12: Изменение кода файла *lab6-3.asm*

```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
10tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
```

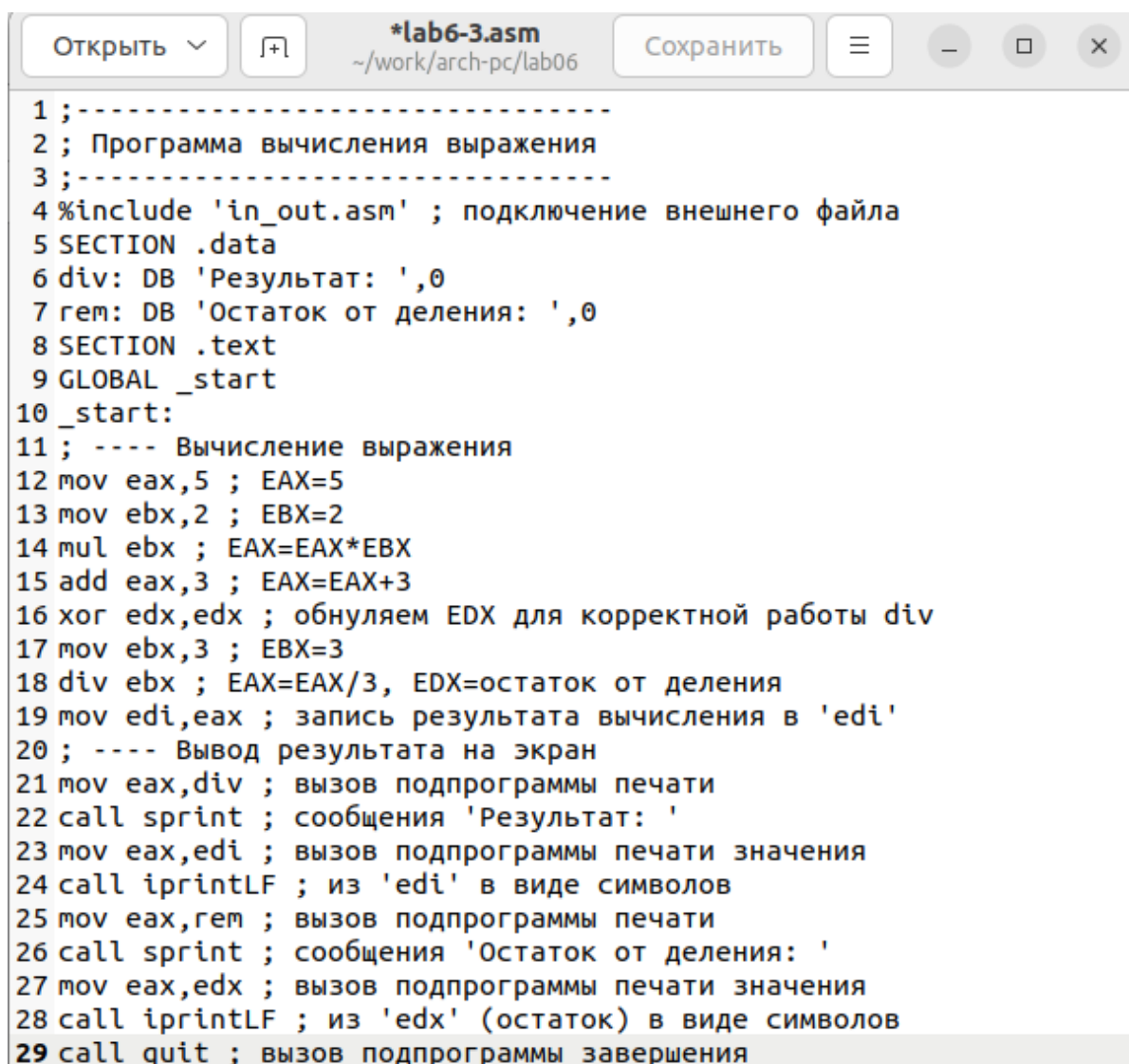
Рис. 2.13: Компиляция изменённого файла *lab6-3.asm* и запуск кода

Создал файл *lab6-3.asm* (рис. 2.14). Ввёл код программы из листинга 6.3 в файл (рис. 2.15). Скомпилировал файл и запустил код для проверки его работы (рис. 2.16).

Внёс необходимые изменения в файл *lab6-3.asm* (рис. 2.17). Скомпилировал его и, запустив, проверил работу (рис. 2.18).

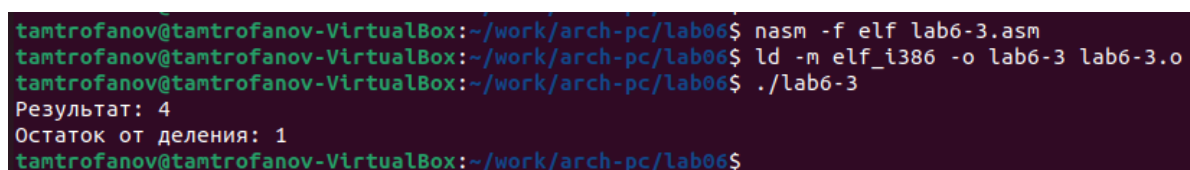
```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.14: Создание файла файл *lab6-3.asm*



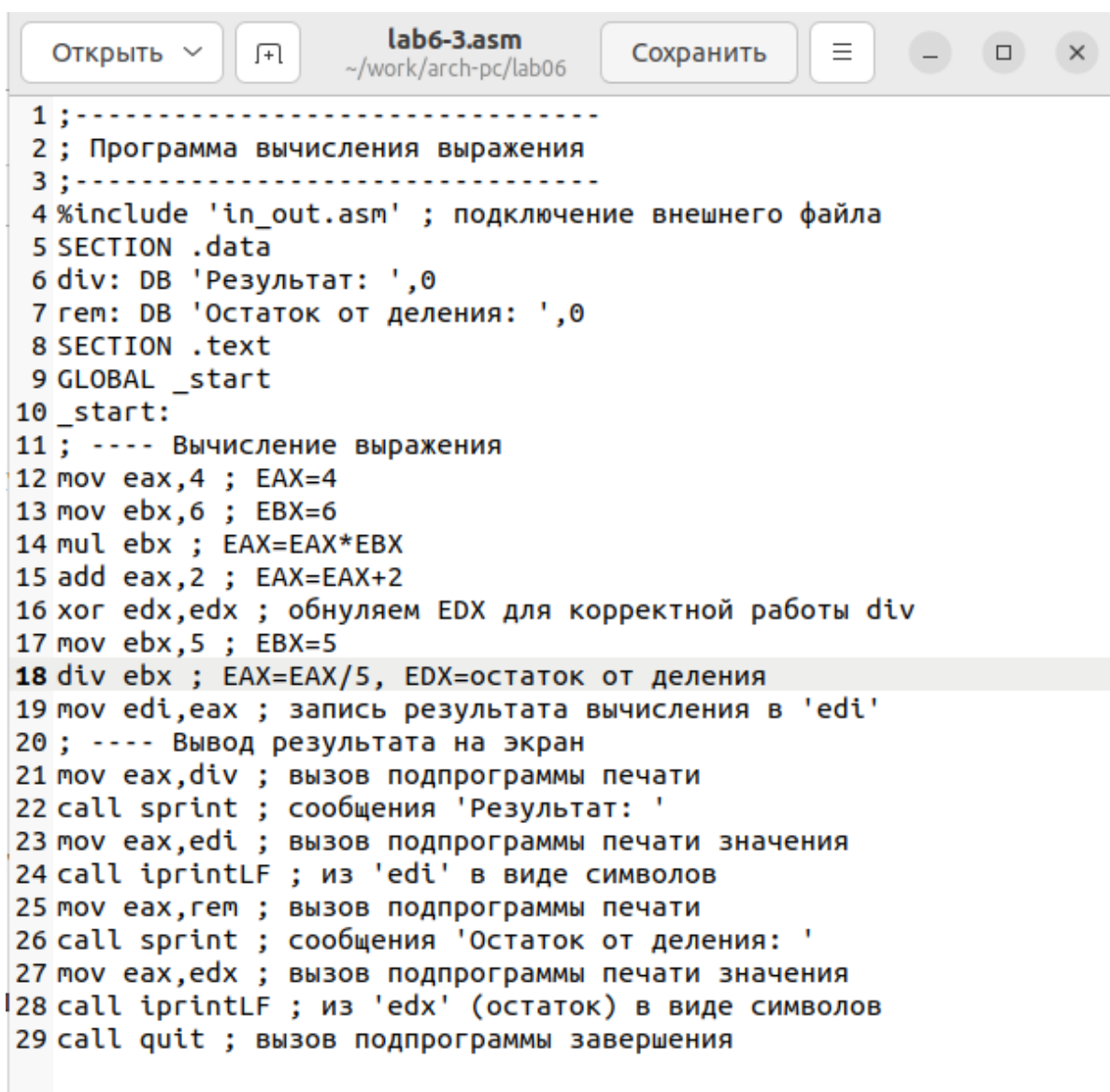
```
1 ;-----
2 ; Программа вычисления выражения
3 ;-----
4 %include 'in_out.asm' ; подключение внешнего файла
5 SECTION .data
6 div: DB 'Результат: ',0
7 rem: DB 'Остаток от деления: ',0
8 SECTION .text
9 GLOBAL _start
10 _start:
11 ; ---- Вычисление выражения
12 mov eax,5 ; EAX=5
13 mov ebx,2 ; EBX=2
14 mul ebx ; EAX=EAX*EBX
15 add eax,3 ; EAX=EAX+3
16 xor edx,edx ; обнуляем EDX для корректной работы div
17 mov ebx,3 ; EBX=3
18 div ebx ; EAX=EAX/3, EDX=остаток от деления
19 mov edi,eax ; запись результата вычисления в 'edi'
20 ; ---- Вывод результата на экран
21 mov eax,div ; вызов подпрограммы печати
22 call sprint ; сообщения 'Результат: '
23 mov eax,edi ; вызов подпрограммы печати значения
24 call iprintLF ; из 'edi' в виде символов
25 mov eax,rem ; вызов подпрограммы печати
26 call sprint ; сообщения 'Остаток от деления: '
27 mov eax,edx ; вызов подпрограммы печати значения
28 call iprintLF ; из 'edx' (остаток) в виде символов
29 call quit ; вызов подпрограммы завершения
```

Рис. 2.15: Внесённый код в файл *lab6-3.asm*



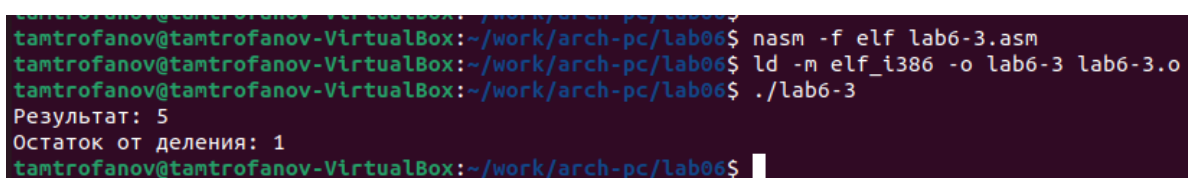
```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.16: Компиляция файла *lab6-3.asm* и запуск кода



```
1 ;-----
2 ; Программа вычисления выражения
3 ;-----
4 %include 'in_out.asm' ; подключение внешнего файла
5 SECTION .data
6 div: DB 'Результат: ',0
7 rem: DB 'Остаток от деления: ',0
8 SECTION .text
9 GLOBAL _start
10 _start:
11 ; ---- Вычисление выражения
12 mov eax,4 ; EAX=4
13 mov ebx,6 ; EBX=6
14 mul ebx ; EAX=EAX*EBX
15 add eax,2 ; EAX=EAX+2
16 xor edx,edx ; обнуляем EDX для корректной работы div
17 mov ebx,5 ; EBX=5
18 div ebx ; EAX=EAX/5, EDX=остаток от деления
19 mov edi,eax ; запись результата вычисления в 'edi'
20 ; ---- Вывод результата на экран
21 mov eax,div ; вызов подпрограммы печати
22 call sprint ; сообщения 'Результат: '
23 mov eax,edi ; вызов подпрограммы печати значения
24 call iprintLF ; из 'edi' в виде символов
25 mov eax,rem ; вызов подпрограммы печати
26 call sprint ; сообщения 'Остаток от деления: '
27 mov eax,edx ; вызов подпрограммы печати значения
28 call iprintLF ; из 'edx' (остаток) в виде символов
29 call quit ; вызов подпрограммы завершения
```

Рис. 2.17: Изменённый файл *lab6-3.asm*



```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
```

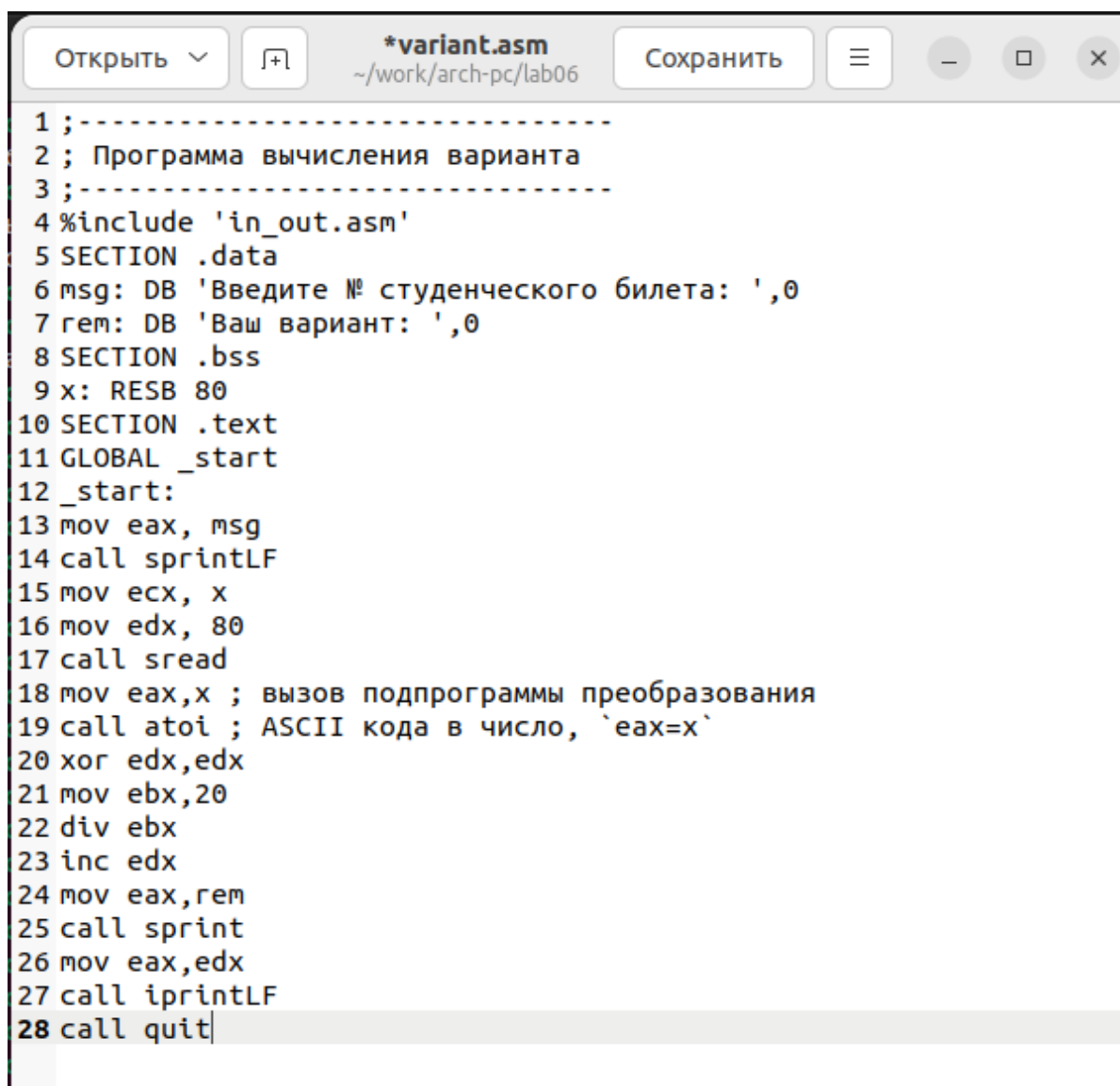
Рис. 2.18: Компиляция изменённого файла *lab6-3.asm* и запуск кода

Создал файл *variant.asm* (рис. 2.19). Внёс код программы листинга 6.4 в файл

(рис. 2.20). Скомпелировал файл и проверил его работу (рис. 2.21). Затем проверил работу файла аналитически (рис. 2.22).

```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$  
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm  
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 2.19: Создание файла файл *variant.asm*



```
1 ;-----  
2 ; Программа вычисления варианта  
3 ;-----  
4 %include 'in_out.asm'  
5 SECTION .data  
6 msg: DB 'Введите № студенческого билета: ',0  
7 rem: DB 'Ваш вариант: ',0  
8 SECTION .bss  
9 x: RESB 80  
10 SECTION .text  
11 GLOBAL _start  
12 _start:  
13 mov eax, msg  
14 call sprintfLF  
15 mov ecx, x  
16 mov edx, 80  
17 call sread  
18 mov eax,x ; вызов подпрограммы преобразования  
19 call atoi ; ASCII кода в число, `eax=x`  
20 xor edx,edx  
21 mov ebx,20  
22 div ebx  
23 inc edx  
24 mov eax,rem  
25 call sprintf  
26 mov eax,edx  
27 call iprintLF  
28 call quit
```

Рис. 2.20: Внесённый код в файл *variant.asm*

```

tamtrofanov@tamtrofanov-VirtualBox: ~/work/arch-pc/lab06$
tamtrofanov@tamtrofanov-VirtualBox: ~/work/arch-pc/lab06$ nasm -f elf variant.asm
tamtrofanov@tamtrofanov-VirtualBox: ~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
tamtrofanov@tamtrofanov-VirtualBox: ~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132231842
Ваш вариант: 3
tamtrofanov@tamtrofanov-VirtualBox: ~/work/arch-pc/lab06$

```

Рис. 2.21: Компиляция файла *variant.asm* и запуск кода

1132231842 mod 20	=	2
2+1	=	3
3		

Рис. 2.22: Аналитическая проверка работы кода

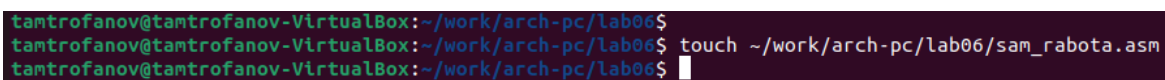
2.1 Ответ на вопросы

- 1) За вывод строки ***Ваш вариант:*** отвечают 2 строки: **`mov eax,rem call sprint`**
- 2) объявляется переменная, под неё выделяется 80 байт, а затем вызывается функция которая отвечает за ввод строки пользователем в ранее объявленную переменную.
- 3) Это функция преобразования ascii-код символа в целое число
- 4) За вычисление варианта отвечает следующая последовательность строк - **`xor edx,edx mov ebx,20 div ebx inc edx`**
- 5) Запись происходит в регистр edx.
- 6) Данная инструкция преобразует 1 к значению регистра edx.
- 7) Строки - **`mov eax,edx call iprintLF`**

отвечают за вывод результата вычислений в консоль

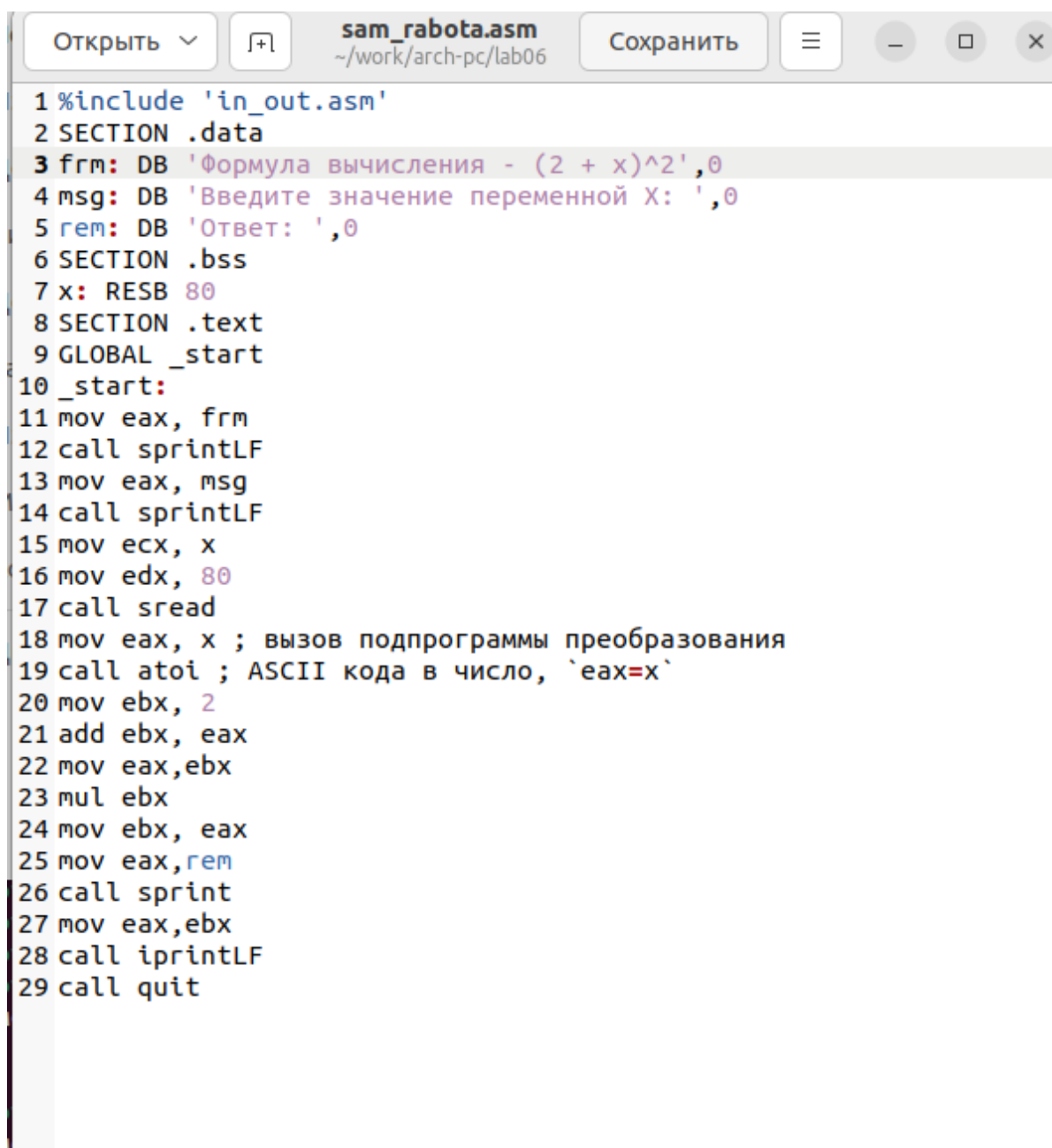
3 Задание для самостоятельной работы

Создаю файл *sam_rabota.asm* (рис. 3.1). Пишу необходимый код (рис. 3.2). Компилирую файл и запускаю его код, проверяю работу программы (рис. 3.3).

A screenshot of a terminal window with a dark background and light-colored text. It shows three lines of text: the first line is the prompt 'tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06\$'; the second line is the command 'touch ~/work/arch-pc/lab06/sam_rabota.asm'; and the third line is the prompt 'tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06\$' followed by a cursor. The text is in a monospaced font.

```
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$  
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/sam_rabota.asm  
tamtrofanov@tamtrofanov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 3.1: Создание файла файл *sam_rabota.asm*



```
1 %include 'in_out.asm'
2 SECTION .data
3 frm: DB 'Формула вычисления - (2 + x)^2',0
4 msg: DB 'Введите значение переменной X: ',0
5 rem: DB 'Ответ: ',0
6 SECTION .bss
7 x: RESB 80
8 SECTION .text
9 GLOBAL _start
10 _start:
11 mov eax, frm
12 call sprintLF
13 mov eax, msg
14 call sprintLF
15 mov ecx, x
16 mov edx, 80
17 call sread
18 mov eax, x ; вызов подпрограммы преобразования
19 call atoi ; ASCII кода в число, `eax=x`
20 mov ebx, 2
21 add ebx, eax
22 mov eax, ebx
23 mul ebx
24 mov ebx, eax
25 mov eax, rem
26 call sprint
27 mov eax, ebx
28 call iprintLF
29 call quit
```

Рис. 3.2: Внесённый код в файл *sam_rabota.asm*

```
tantrofanov@tantrofanov-VirtualBox:~/work/arch-pc/lab06$  
tantrofanov@tantrofanov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf sam_rabota.asm  
tantrofanov@tantrofanov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o sam_rabota sam_rabota.o  
tantrofanov@tantrofanov-VirtualBox:~/work/arch-pc/lab06$ ./sam_rabota  
Формула вычисления -  $(2 + x)^2$   
Введите значение переменной X:  
2  
Ответ: 16  
tantrofanov@tantrofanov-VirtualBox:~/work/arch-pc/lab06$ ./sam_rabota  
Формула вычисления -  $(2 + x)^2$   
Введите значение переменной X:  
8  
Ответ: 100  
tantrofanov@tantrofanov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 3.3: Компиляция файла *sam_rabota.asm*, запуск кода и проверка кода

4 Выводы

Сегодня я освоил арифметические инструкции языка ассемблера NASM.