

Отчёт по Лабораторной работе №6

Дисциплина: Архитектура компьютера

Гозенко А.С.

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выполнение задания для самостоятельной работы.	14
4	Выводы	16

Список иллюстраций

2.1	Создание каталога и файла в нём	5
2.2	Ввод программы из листинга	5
2.3	Запуск исполнительного файла	6
2.4	Изменение файла	6
2.5	Запуск исполнительного файла	6
2.6	Таблица ASCII	7
2.7	Ввод программы	7
2.8	Запуск исполнительного файла	7
2.9	Изменение файла	8
2.10	Запуск исполнительного файла	8
2.11	Запуск исполнительного файла	9
2.12	Ввод кода программы	10
2.13	Запуск исполнительного файла	10
2.14	Изменение файла	11
2.15	Запуск исполнительного файла	11
2.16	Изменение файла	12
2.17	Запуск исполнительного файла	12
3.1	Написание программы	14
3.2	Запуск исполнительного файла	15

1 Цель работы

Цель данной работы - освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Создание каталога, переход в него и создание файла lab6-1.asm (рис. 2.1).

```
asgozenko@dk8n61 ~ $ mkdir ~/work/arch-pc/lab06
asgozenko@dk8n61 ~ $ cd ~/work/arch-pc/lab06
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ touch lab6-1.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $
```

Рис. 2.1: Создание каталога и файла в нём

Ввод в файл lab6-1.asm программы из листинга 6.1 (рис. 2.2).

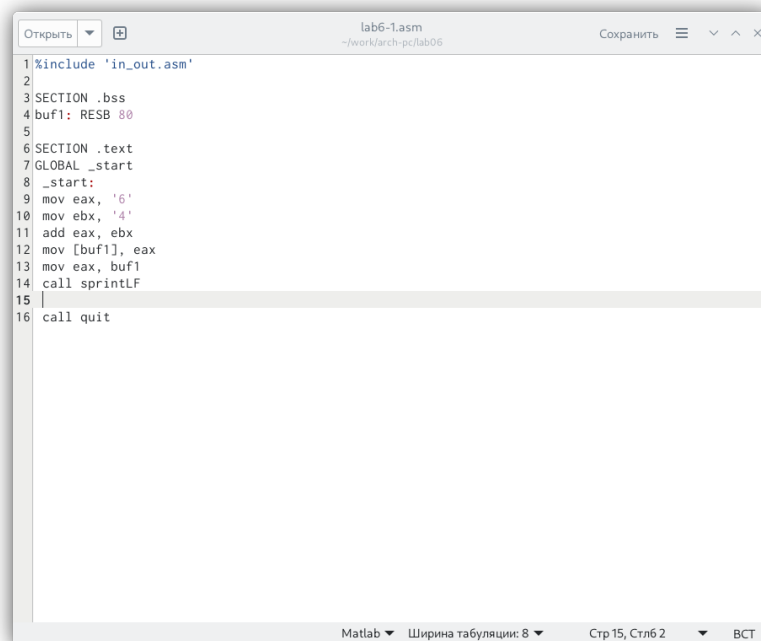


Рис. 2.2: Ввод программы из листинга

Создание исполнительного файла и его запуск (рис. 2.3).

```

asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-1
j
asgozenko@dk8n61 ~/work/arch-pc/lab06 $

```

Рис. 2.3: Запуск исполнительного файла

Изменение текста программы (рис. 2.4).

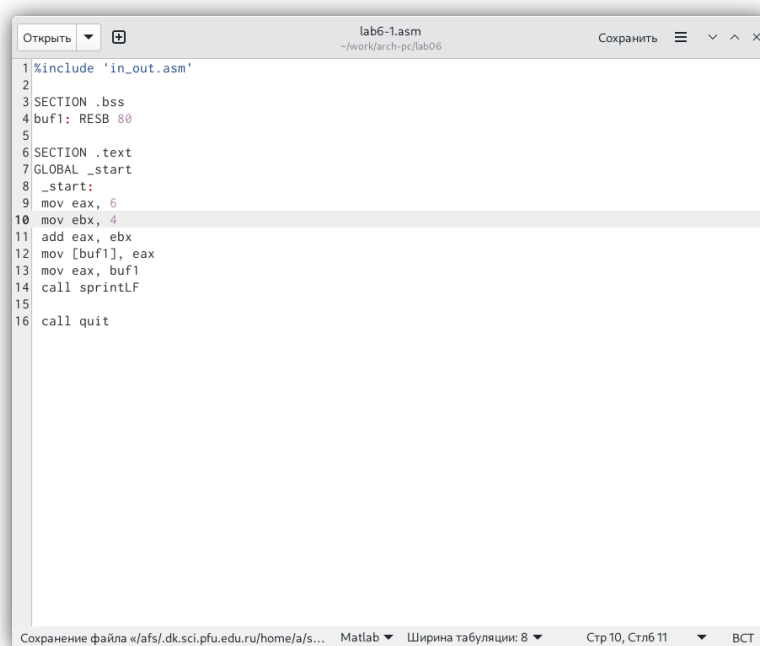


Рис. 2.4: Изменение файла

Создание исполнительного файла и его запуск (рис. 2.5).

```

asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-1
j
asgozenko@dk8n61 ~/work/arch-pc/lab06 $

```

Рис. 2.5: Запуск исполнительного файла

Определение выведенного программой символа в таблице ASCII (рис. 2.6).

9	11	0x09	1001	HT, \t
10	12	0x0A	1010	LF, \n
11	13	0x0B	1011	VT
12	14	0x0C	1100	FF
13	15	0x0D	1101	CR, \r

Рис. 2.6: Таблица ASCII

Ввод программы из листинга 6.2 в файл lab6-2.asm (рис. 2.7).

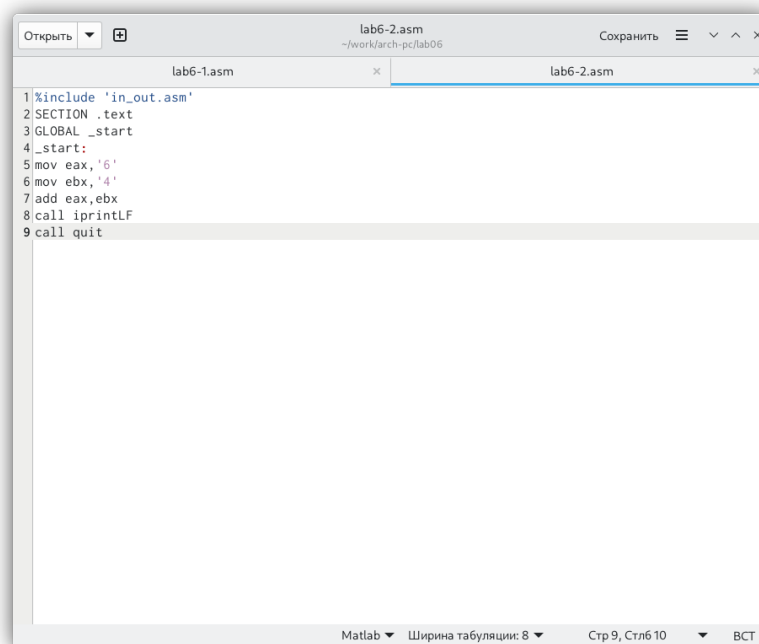


Рис. 2.7: Ввод программы

Создание файла lab6-2.asm, его исполнительного файла и запуск (рис. 2.8).

```

asgozenko@dk8n61 ~/work/arch-pc/lab06 $ touch lab6-2.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-2
106
asgozenko@dk8n61 ~/work/arch-pc/lab06 $
  
```

Рис. 2.8: Запуск исполнительного файла

Изменение файла lab6-2.asm(рис. 2.9).

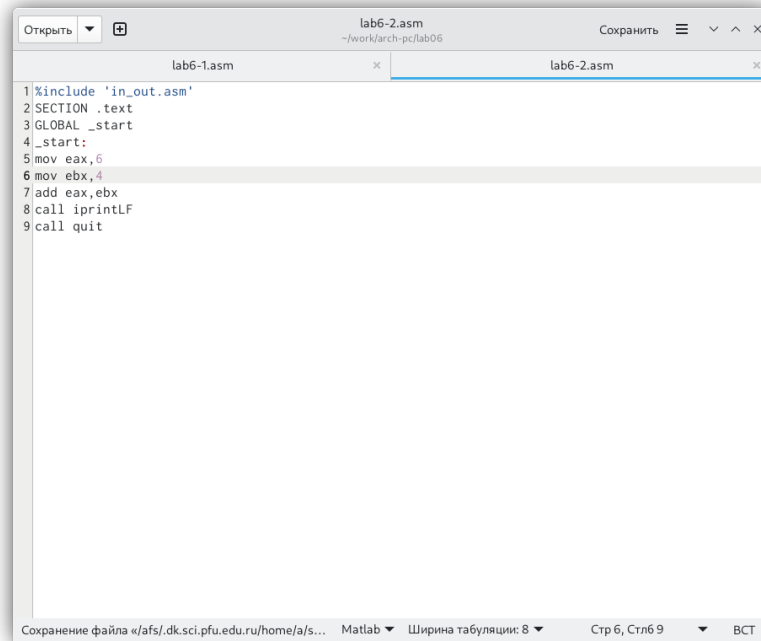


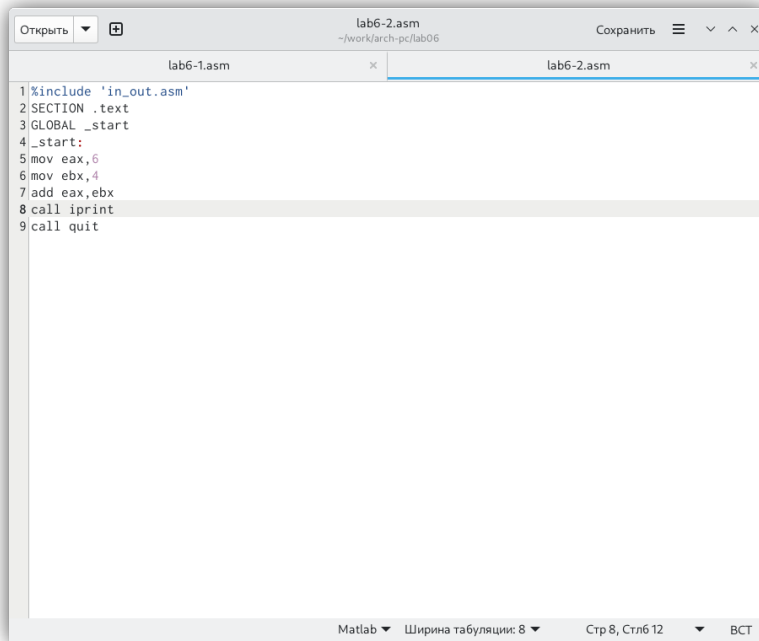
Рис. 2.9: Изменение файла

Создание исполнительного файла и его запуск (рис. 2.10).

```
106
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-2
10
asgozenko@dk8n61 ~/work/arch-pc/lab06 $
```

Рис. 2.10: Запуск исполнительного файла

Изменение файла lab6-2.asm(рис. ??).



Изменился вывод ин-

формации, теперь нет отступа строки.

Создание исполняемого файла и его запуск(рис. 2.11).

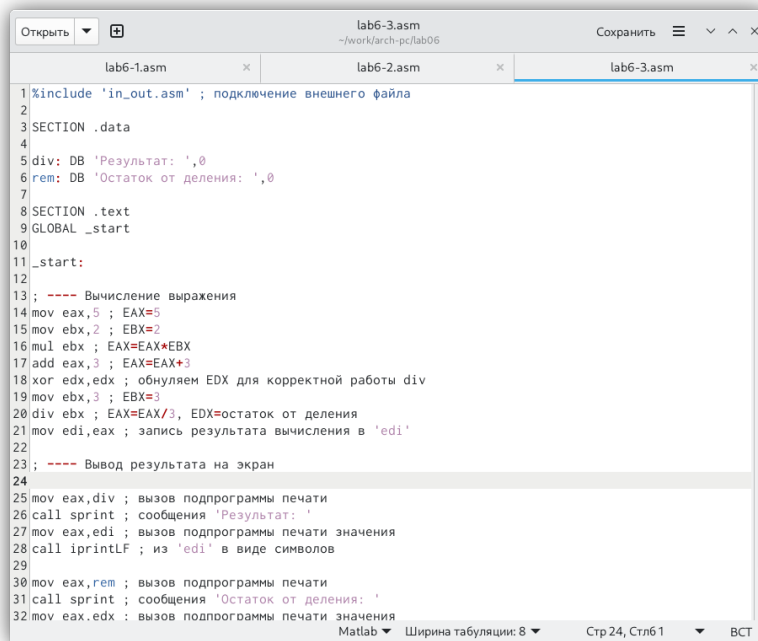
```

asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-2
10
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-2
10asgozenko@dk8n61 ~/work/arch-pc/lab06 $

```

Рис. 2.11: Запуск исполняемого файла

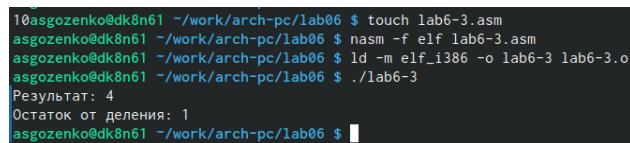
Ввод программы из листинга 6.3 в файл lab6-3.asm (рис. 2.12).



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

Рис. 2.12: Ввод кода программы

Создание исполнительного файла и его запуск(рис. 2.13).



```
10asgozenko@dk8n61 ~/work/arch-pc/lab06 $ touch lab6-3.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
asgozenko@dk8n61 ~/work/arch-pc/lab06 $
```

Рис. 2.13: Запуск исполнительного файла

Изменение файла lab6-3.asm(рис. 2.14).

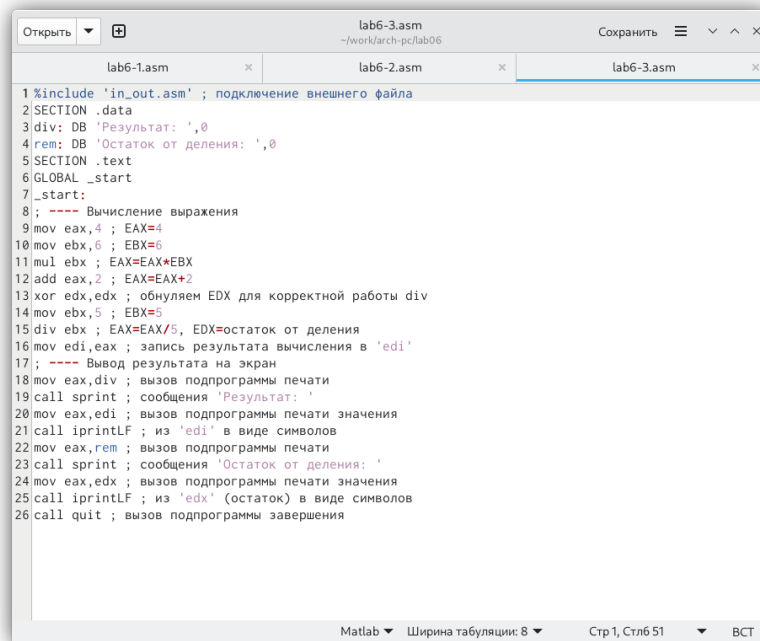


Рис. 2.14: Изменение файла

Создание исполнительного файла и его запуск(рис. 2.15).

```

Остаток от деления: 2
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
asgozenko@dk8n61 ~/work/arch-pc/lab06 $

```

Рис. 2.15: Запуск исполнительного файла

Изменение файла variant.asm(рис. 2.16).

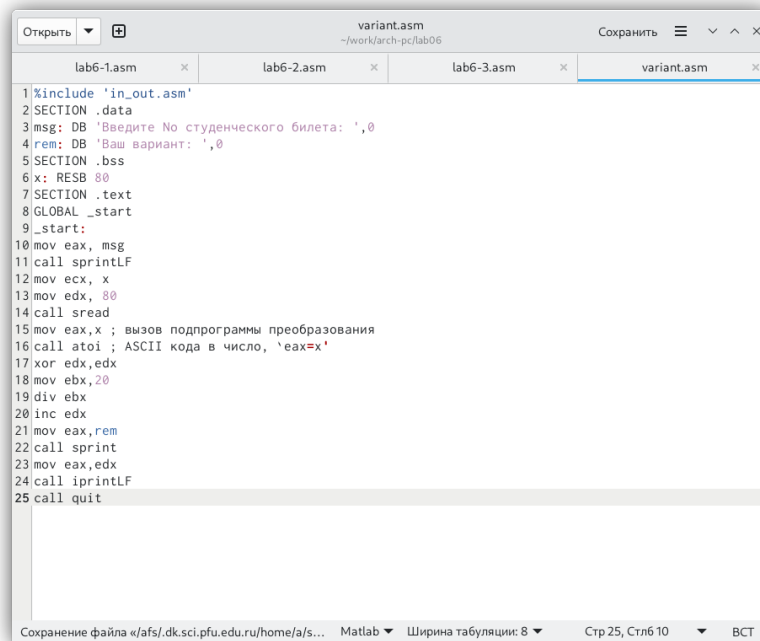


Рис. 2.16: Изменение файла

Создание исполнительного файла и его запуск(рис. 2.17).

```

asgozenko@dk8n61 ~/work/arch-pc/lab06 $ touch variant.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf variant3.asm
nasm: fatal: unable to open input file 'variant3.asm' No such file or directory
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета:
1132236112
Ваш вариант: 13
asgozenko@dk8n61 ~/work/arch-pc/lab06 $

```

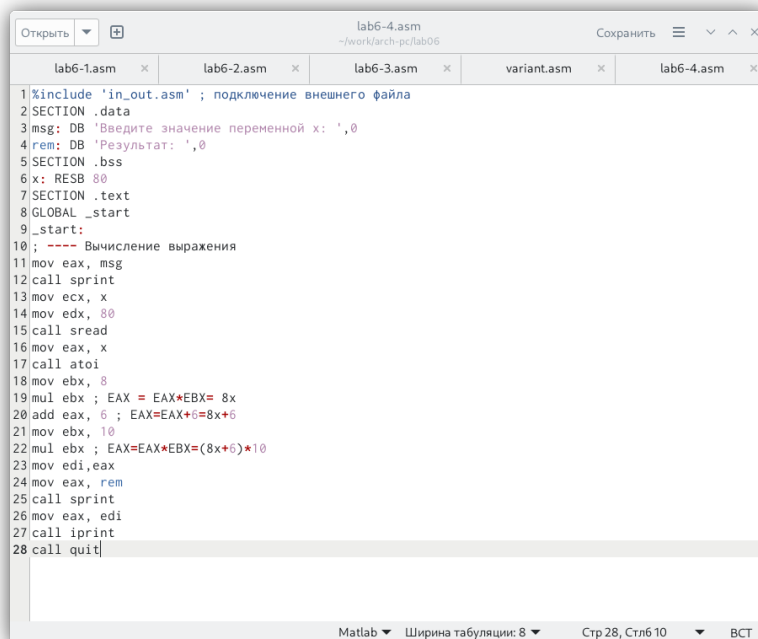
Рис. 2.17: Запуск исполнительного файла

Ответы на вопросы: 1. За вывод сообщение “Ваш вариант” отвечают строки: `mov eax, rem` `call sprint` 2. Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающий ввод сообщения с клавиатуры. 3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`. 4. За вычисление варианта

отвечают строки: `xor edx,edx mov ebx,20 div ebx inc edx` 5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`. 6. инструкция `inc edx` увеличивает значение регистра `edx` на 1. 7. За вывод на экран результатов вычислений отвечают строки: `mov eax,edx call iprintLF`

3 Выполнение задания для самостоятельной работы.

Написание программы для вычисления значения функции $(8x + 6) \times 10$ (рис. 3.1).



```
lab6-4.asm
~/work/arch-pc/lab06
Сохранить

lab6-1.asm x lab6-2.asm x lab6-3.asm x variant.asm x lab6-4.asm x

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg: DB 'Введите значение переменной x: ',0
4 rem: DB 'Результат: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 ; ---- Вычисление выражения
11 mov eax, msg
12 call sprint
13 mov ecx, x
14 mov edx, 80
15 call sread
16 mov eax, x
17 call atoi
18 mov ebx, 8
19 mul ebx ; EAX = EAX*EBX = 8x
20 add eax, 6 ; EAX=EAX+6=8x+6
21 mov ebx, 10
22 mul ebx ; EAX=EAX*EBX=(8x+6)*10
23 mov edi,eax
24 mov eax, rem
25 call sprint
26 mov eax, edi
27 call iprint
28 call quit

Matlab Ширина табуляции: 8 Стр 28, Стлб 10 ВСТ
```

Рис. 3.1: Написание программы

Создание исполнительного файла и его запуск(рис. 3.2).

```

asgozenko@dk8n61 ~/work/arch-pc/lab06 $ touch lab6-4.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
lab6-4.asm:23: error: parser: instruction expected
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
lab6-4.asm:23: error: parser: instruction expected
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
lab6-4.asm:24: error: symbol 'rem' not defined
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x: 1
Результат: 140asgozenko@dk8n61 ~/work/arch-pc/lab06 $ ./lab6-4
Введите значение переменной x: 4
Результат: 380asgozenko@dk8n61 ~/work/arch-pc/lab06 $

```

Рис. 3.2: Запуск исполнительного файла

Код программы: `%include 'in_out.asm'` ; подключение внешнего файла
`SECTION .data msg: DB 'Введите значение переменной x:',0 rem: DB 'Результат:',0`
`SECTION .bss x: RESB 80`
`SECTION .text GLOBAL _start _start: ;` — Вычисление выражения
`mov eax, msg call sprint mov ecx, x mov edx, 80 call sread mov eax, x call atoi mov`
`ebx, 8 mul ebx ; $EAX = EAX \cdot EBX = 8x$ add eax, 6 ; $EAX = EAX + 6 = 8x + 6$ mov ebx, 10 mul ebx ;`
 `$EAX = EAX \cdot EBX = (8x + 6) \cdot 10$ mov edi, eax mov eax, rem call sprint mov eax, edi call iprint`
`call quit`

4 Выводы

При выполнении данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.