

Арифметические операции в NASM

Лабораторная работа №6

Владимир Романович Козомазов

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Описание выполняемого задания	6
2.2	Выполнение заданий лабораторной работы	6
2.2.1	Символьные и численные данные в NASM	6
2.2.2	Выполнение арифметических операций в NASM	10
2.2.3	Ответы на вопросы по заданию:	13
2.3	Выводы по результатам выполнения заданий	14
3	Выполнение самостоятельного задания	16
3.1	Описание выполняемого самостоятельного задания	16
3.2	Выполнение заданий для самостоятельной работы	16
3.3	Выводы по результатам выполнения самостоятельного задания . . .	18
3.4	Листинги написанных программ	18
4	Выводы	24

Список иллюстраций

2.1	Создание директории для программ лабораторной работы	6
2.2	Ввод текста из листинга 6.1 в файл lab6-1.asm	7
2.3	Трансляция и запуск программы lab6-1	7
2.4	Изменения кода в файле lab6-1.asm	8
2.5	Повторный запуск программы lab6-1	8
2.6	Ввод кода программы из листинга 6.2 в файл lab6-2	8
2.7	Изменения кода в файле lab6-2	9
2.8	Повторный запуск программы lab6-2	9
2.9	Замена функции iprintLF на iprint в файле lab6-2	10
2.10	Запуск программы lab6-2 с функцией iprint вместо iprintLF	10
2.11	Ввод кода из листинга 6.3 в файл lab6-3.asm	11
2.12	Запуск программы lab6-3	11
2.13	Изменение текста программы в файле 'lab6-3.asm'	12
2.14	Запуск программы 'lab6-3'	12
2.15	Создание нового файла variant.asm и ввод кода из листинга 6.4 . .	13
2.16	Запуск программы variant и вывод результата работы на экран . .	13
3.1	Создание файла calc_function.asm и написание кода для выполне- ния задания	17
3.2	Результаты запуска программы calc_function и вывод результата работы на экран	17

Список таблиц

1 Цель работы

- Изучить работу с численными и символьными данными в языке ассемблера NASM.
- Изучить арифметические операции в языке ассемблера NASM и поработать с ними на практике.

2 Выполнение лабораторной работы

2.1 Описание выполняемого задания

- Написать программы для работы с численными и символьными данными и выводом их на экран.
- Написать программы для работы с арифметическими операциями и вычисления значения арифметического выражения.

2.2 Выполнение заданий лабораторной работы

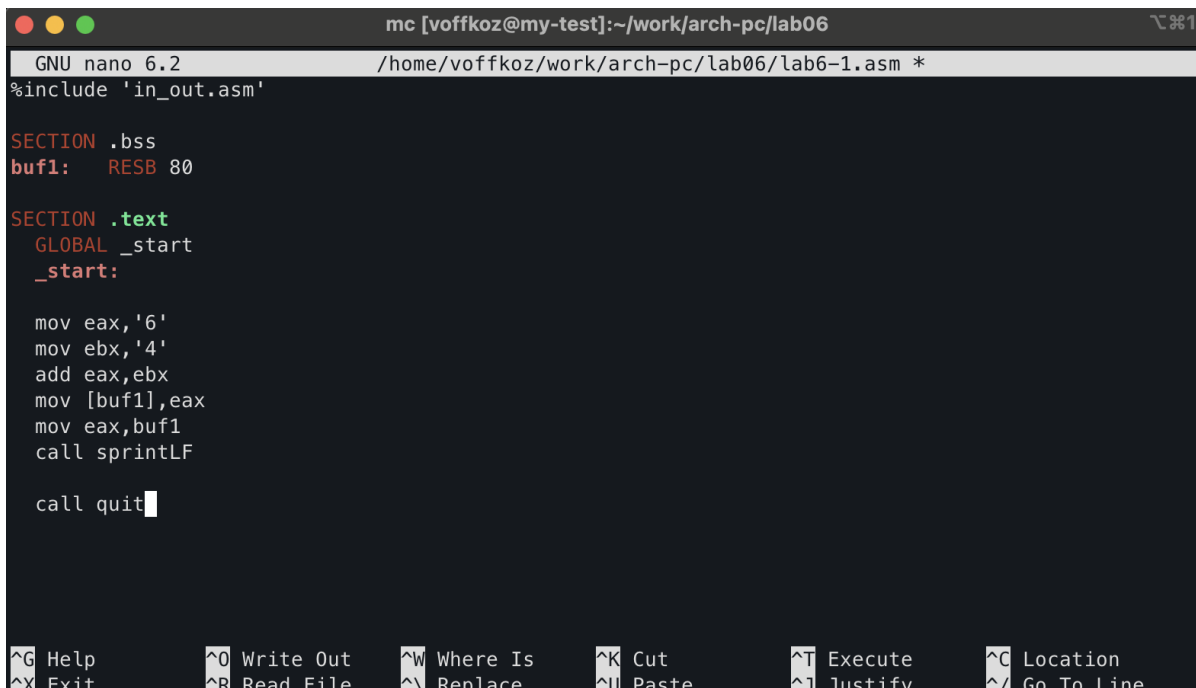
2.2.1 Символьные и численные данные в NASM

Создал каталог для программ лабораторной работы №6, перешёл в него и создал файл `lab6-1.asm` (рис. 2.1)

```
> cd ~/work/arch-pc/  
> mkdir ~/work/arch-pc/lab06  
> cd lab06  
> touch lab6-1.asm
```

Рис. 2.1: Создание директории для программ лабораторной работы

Ввёл в файл `lab6-1.asm` текст программы из листинга 6.1 (рис. 2.2)



```
mc [voffkoz@my-test]:~/work/arch-pc/lab06
GNU nano 6.2 /home/voffkoz/work/arch-pc/lab06/lab6-1.asm *
#include 'in_out.asm'

SECTION .bss
buf1:  RESB 80

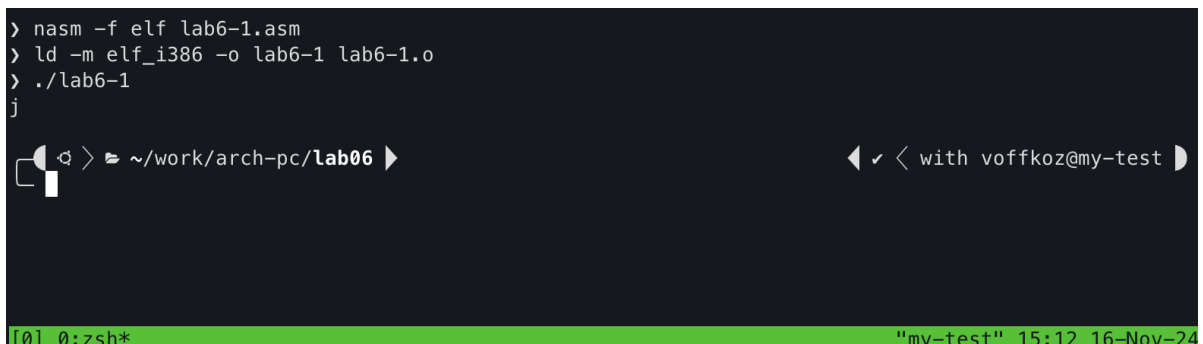
SECTION .text
GLOBAL _start
_start:

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

    call quit
```

Рис. 2.2: Ввод текста из листинга 6.1 в файл lab6-1.asm

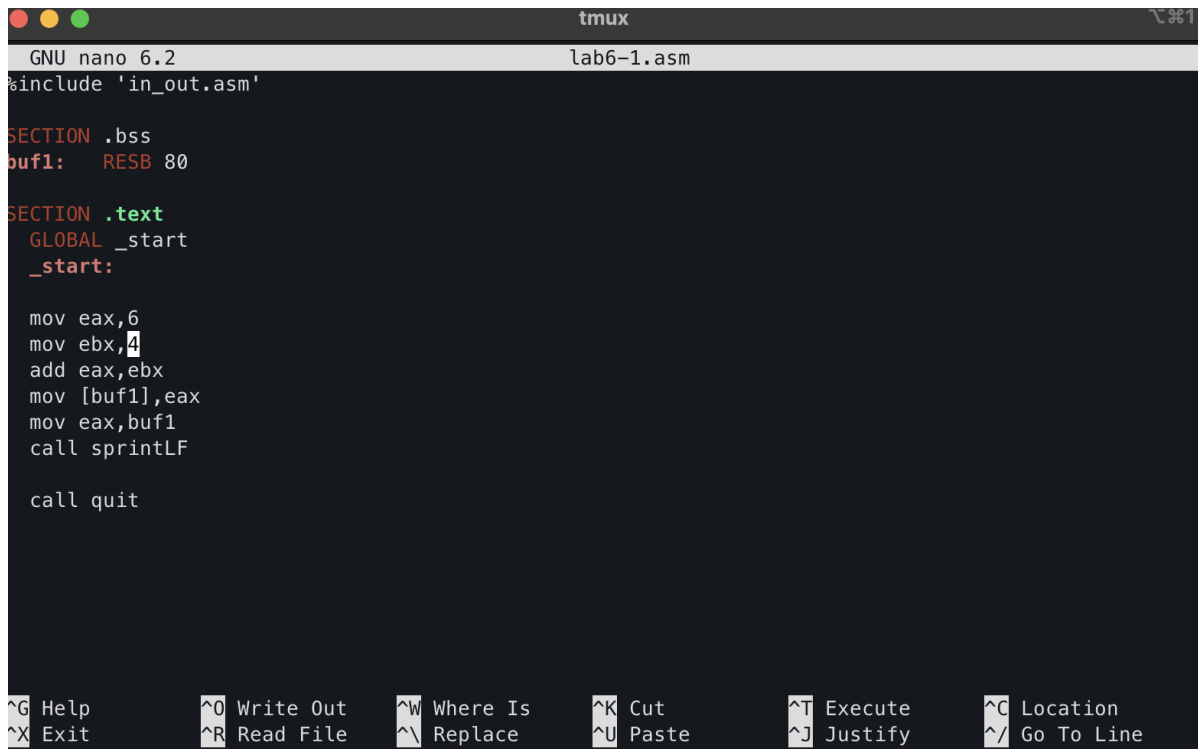
Создал исполняемый файл и запустил его (рис. 2.3)



```
> nasm -f elf lab6-1.asm
> ld -m elf_i386 -o lab6-1 lab6-1.o
> ./lab6-1
j
~ /work/arch-pc/lab06
with voffkoz@my-test
```

Рис. 2.3: Трансляция и запуск программы lab6-1

Изменил текст программы, вместо символов подставил числа (рис. 2.4)



```
GNU nano 6.2 lab6-1.asm
#include 'in_out.asm'

SECTION .bss
buf1:  RESB 80

SECTION .text
GLOBAL _start
_start:

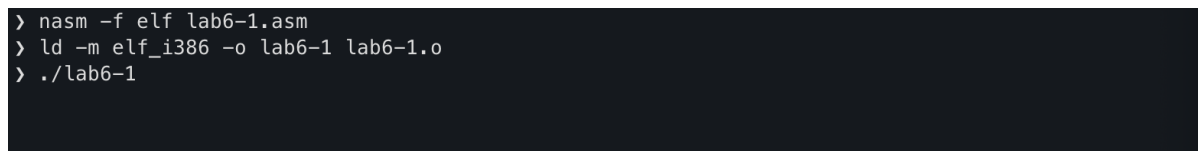
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintf

call quit
```

Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Line

Рис. 2.4: Изменения кода в файле lab6-1.asm

Повторно создал исполняемый файл и запустил его (рис. 2.5). В результате выполнения программы на экран вывелась пустая строка, что говорит о том, что на экран вывелся символ с кодом 10 - LF.



```
> nasm -f elf lab6-1.asm
> ld -m elf_i386 -o lab6-1 lab6-1.o
> ./lab6-1
```

Рис. 2.5: Повторный запуск программы lab6-1

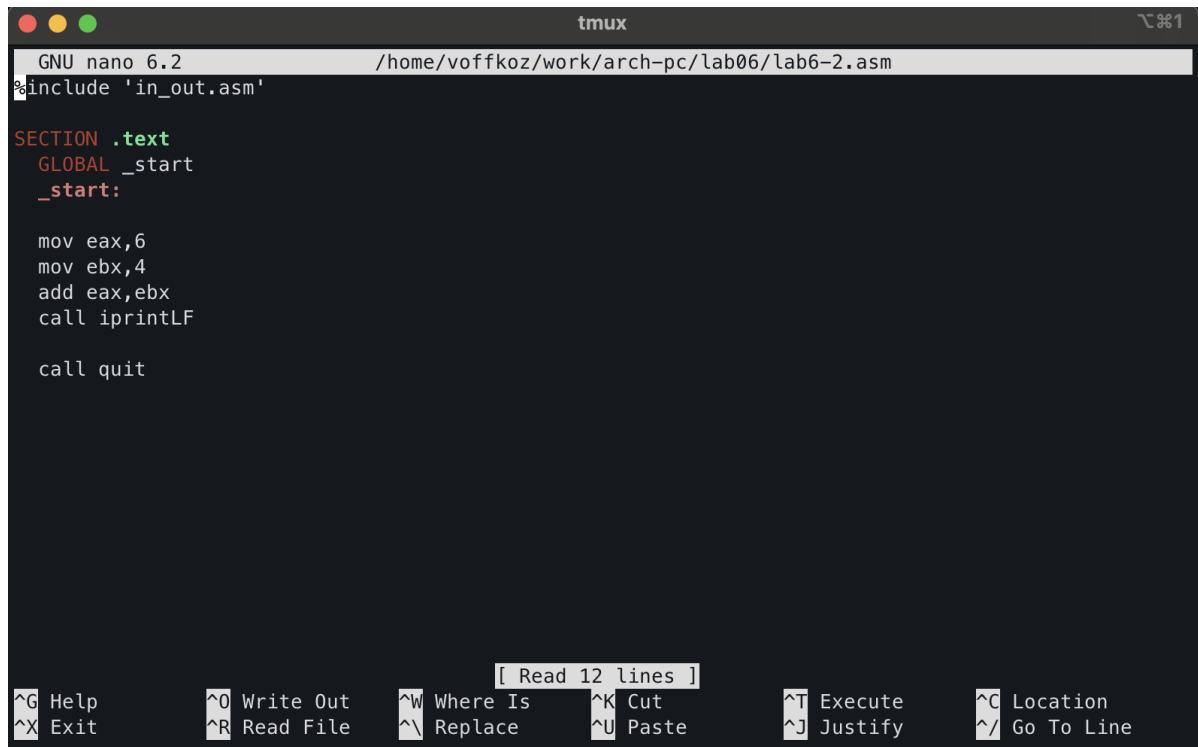
Создал файл lab6-2.asm и ввёл в него код из листинга 6.2 (рис. 2.6)



```
> nano lab6-2.asm
> nasm -f elf lab6-2.asm
> ld -m elf_i386 -o lab6-2 lab6-2.o
> ./lab6-2
106
```

Рис. 2.6: Ввод кода программы из листинга 6.2 в файл lab6-2

Аналогично предыдущему примеру изменил текст программы lab6-2.asm, заменив символы на числа (рис. 2.7)



```
GNU nano 6.2 /home/voffkoz/work/arch-pc/lab06/lab6-2.asm
%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: Изменения кода в файле lab6-2

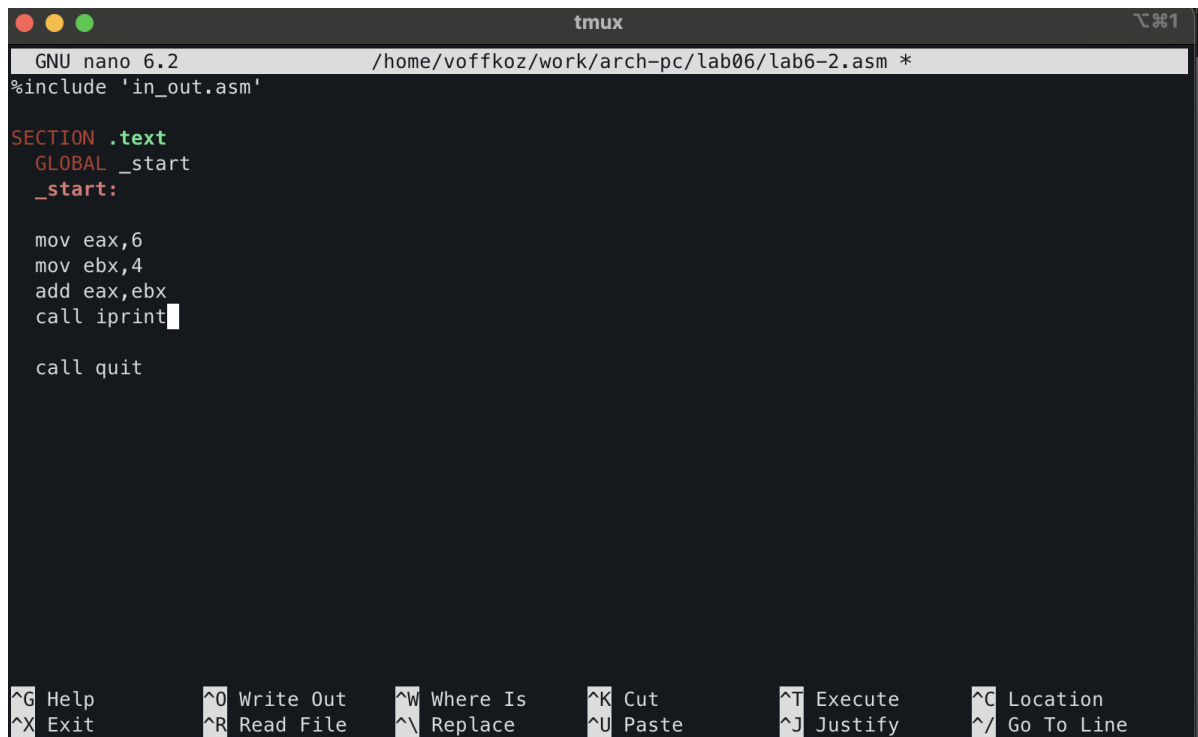
Повторно скомпилировал и запустил программу lab6-2 (рис. 2.8). В результате получили, как и ожидалось, число 10



```
> nasm -f elf lab6-2.asm
> ld -m elf_i386 -o lab6-2 lab6-2.o
> ./lab6-2
10
```

Рис. 2.8: Повторный запуск программы lab6-2

Заменил функцию iprintLF на iprint (рис. 2.9)



```
GNU nano 6.2 /home/voffkoz/work/arch-pc/lab06/lab6-2.asm *
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

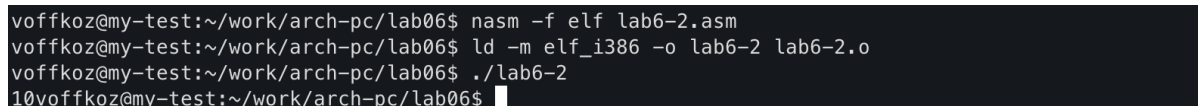
mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit
```

Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line

Рис. 2.9: Замена функции `iprintLF` на `iprint` в файле `lab6-2`

Повторно скомпилировал и запустил программу `lab6-2` (рис. 2.10). Вывод программы отличается тем, что после её выполнения не происходит переход на новую строку.

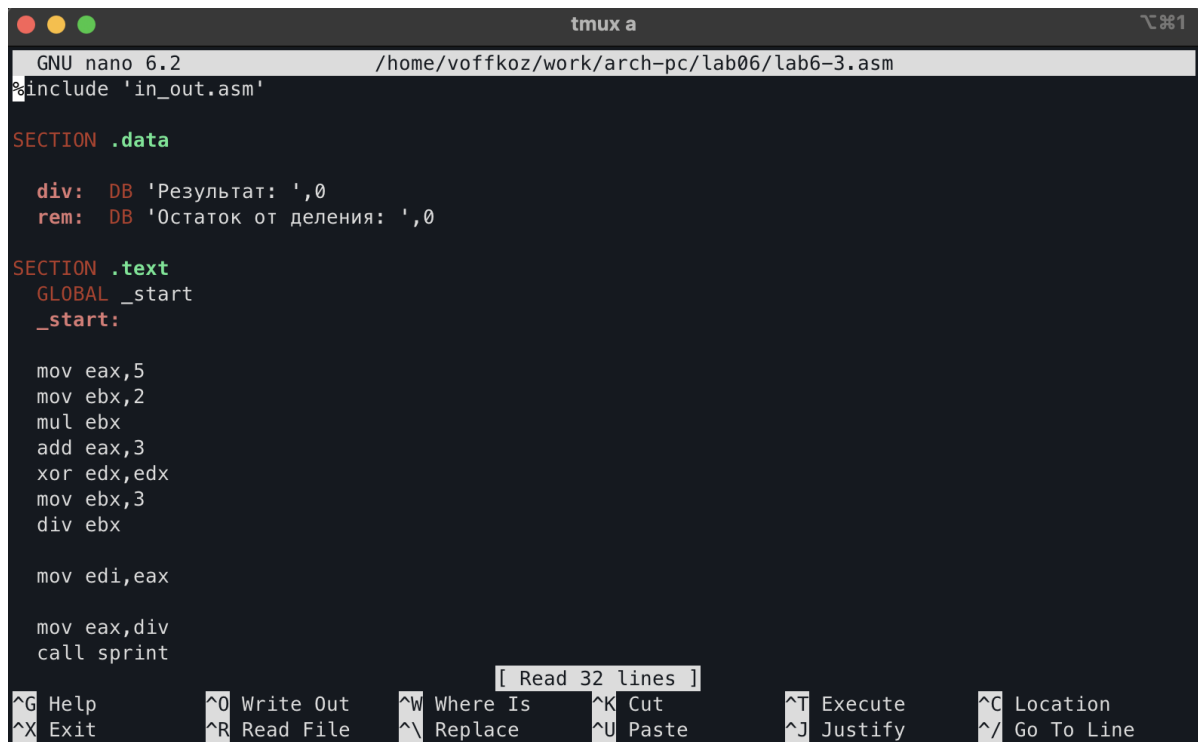


```
voffkoz@my-test:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
voffkoz@my-test:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
voffkoz@my-test:~/work/arch-pc/lab06$ ./lab6-2
10voffkoz@my-test:~/work/arch-pc/lab06$
```

Рис. 2.10: Запуск программы `lab6-2` с функцией `iprint` вместо `iprintLF`

2.2.2 Выполнение арифметических операций в NASM

Создал файл `lab6-3.asm` и ввёл в него код из листинга 6.3 (рис. 2.11)



```
GNU nano 6.2 /home/voffkoz/work/arch-pc/lab06/lab6-3.asm
%include 'in_out.asm'

SECTION .data

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

SECTION .text
GLOBAL _start
_start:

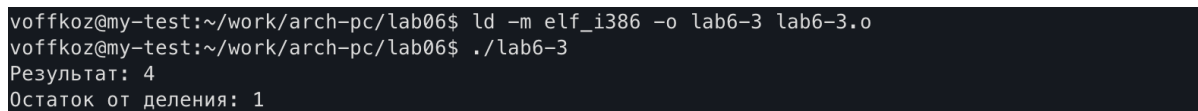
    mov eax,5
    mov ebx,2
    mul ebx
    add eax,3
    xor edx,edx
    mov ebx,3
    div ebx

    mov edi,eax

    mov eax,div
    call sprint
```

Рис. 2.11: Ввод кода из листинга 6.3 в файл lab6-3.asm

Скомпилировал и запустил программу lab6-3 (рис. 2.12)



```
voffkoz@my-test:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
voffkoz@my-test:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 2.12: Запуск программы lab6-3

Изменил текст файла lab6-3.asm так, чтобы он вычислял значение выражения $(4 * 6 + 2)/5$ (рис. 2.13)



```
tmux a
GNU nano 6.2 lab6-3.asm

%include 'in_out.asm'

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

SECTION .text
    GLOBAL _start
    _start:

    mov eax,4
    mov ebx,6
    mul ebx
    add eax,2
    xor edx,edx
    mov ebx,5
    div ebx

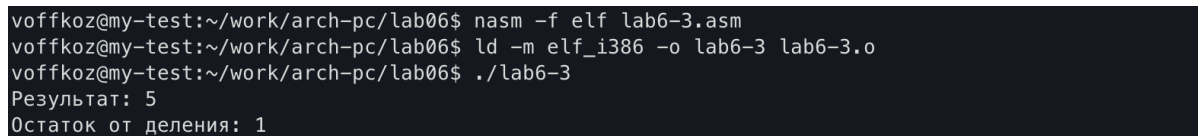
    mov edi,eax

    mov eax,div

[ Read 33 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Рис. 2.13: Изменение текста программы в файле 'lab6-3.asm'

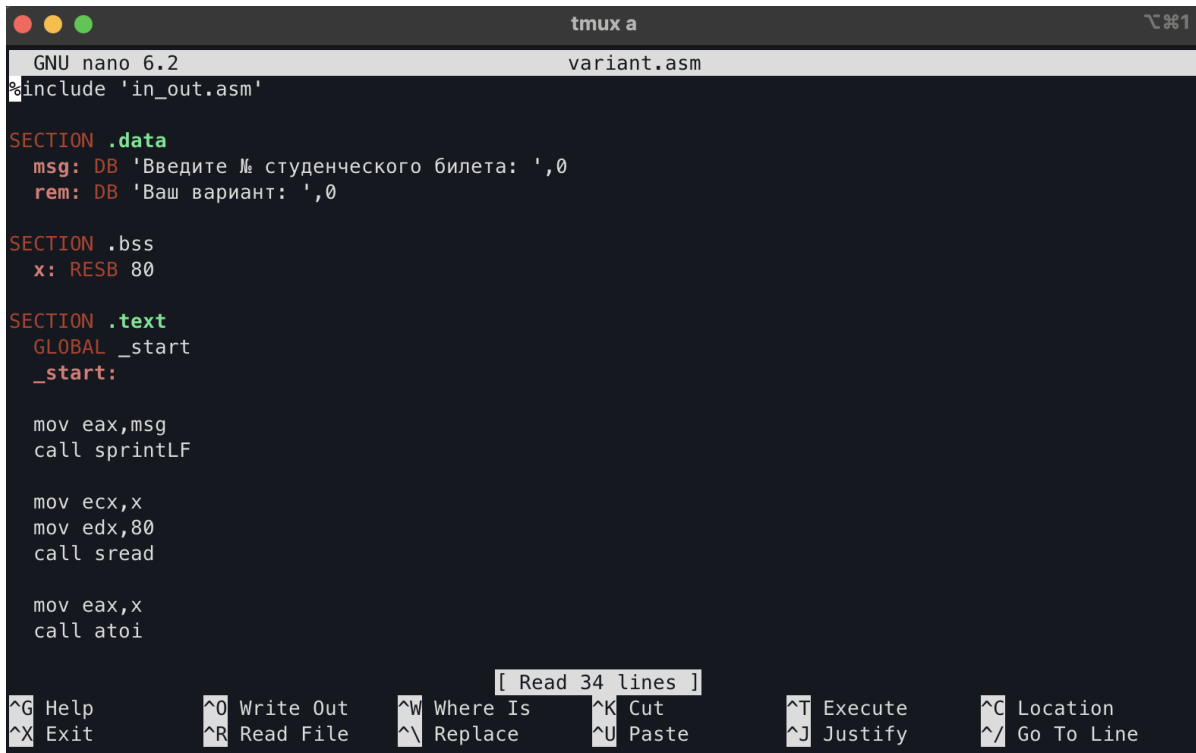
Повторно скомпилировал и запустил программу lab6-3 (рис. 2.14)



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

Рис. 2.14: Запуск программы 'lab6-3'

Для написания программы вычисления варианта задания по номеру студенческого билета создал файл variant.asm и ввёл в него код из листинга 6.4 (рис. 2.15)



```
tmux a  ~%1
GNU nano 6.2 variant.asm
#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 sprintLF

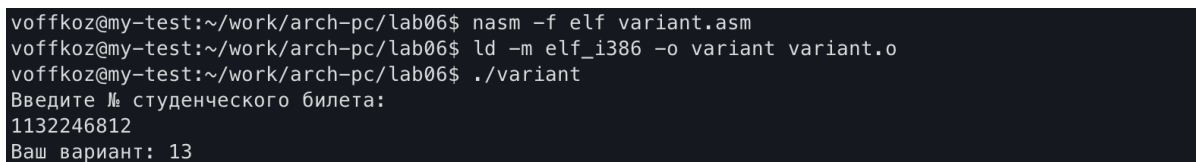
mov ecx,x
mov edx,80
call sread

mov eax,x
call atoi

[ Read 34 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Рис. 2.15: Создание нового файла `variant.asm` и ввод кода из листинга 6.4

Скомпилировал и запустил программу `variant` (рис. 2.16)



```
voffkoz@my-test:~/work/arch-pc/lab06$ nasm -f elf variant.asm
voffkoz@my-test:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
voffkoz@my-test:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246812
Ваш вариант: 13
```

Рис. 2.16: Запуск программы `variant` и вывод результата работы на экран

2.2.3 Ответы на вопросы по заданию:

1. За вывод на экран сообщения “Ваш вариант:” отвечают следующие строки:

```
mov eax,rem
call sprint
```

2. Инструкции

```
mov ecx, x
mov edx, 80
call sread
```

используются для запроса у пользователя данных, которые он вводит с клавиатуры и сохранения их в переменную x.

3. Инструкция `call atoi` используется для преобразования значения, которое ввёл пользователь с клавиатуры из строки в число, с которыми можно производить арифметические операции.
4. За вычисление варианта отвечают следующие строки:

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

5. Остаток от деления при выполнении инструкции `div ebx` записывается в регистр `edx`.
6. Инструкция `inc edx` используется для того, чтобы увеличить остаток от деления на единицу.
7. За вывод на экран результата вычислений отвечают следующие строки:

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

2.3 Выводы по результатам выполнения заданий

В ходе выполнения заданий лабораторной работы были созданы программы для сложения символов и чисел и вывода их на экран. Была продемонстрирована

разница между работой с символами и числами, а также были использованы различные функции для вывода их на экран, в частности такие функции, как `iprint`, `iprintLF` и `atoi`. Была продемонстрирована разница между выводом функций `iprintLF` и `iprint`: вывод функция `iprint` отличается от функции `iprintLF` тем, что после её выполнения не происходит переход на новую строку.

Также в ходе выполнения лабораторной работы был приобретён навык вычисления арифметических выражений, а также освоен метод деления с остатком, который очень часто используется в программировании.

3 Выполнение самостоятельного задания

3.1 Описание выполняемого самостоятельного задания

Написать программу для вычисления выражения $y = f(x)$. Программа должна выводить выражения для вычисления, выводить запрос на ввод значения, вычислять заданное выражение в зависимости от введённого x , выводить результат вычислений.

3.2 Выполнение заданий для самостоятельной работы

Создал файл `calc_function.asm` и написал код для выполнения задания (рис. 3.1)


```
tmux a 18:41
1 %include 'in_out.asm'
2
3 SECTION .data
4     msg:    DB 'Вычисление значения функции (8x + 6) * 10',0
5     request: DB 'Введите значение x: ',0
6     answer: DB 'Результат: ',0
7
8 SECTION .bss
9     x:      RESB 80 ; Значение, которое вводит пользователь
10    buf1:    RESB 80 ; Промежуточная переменная для хранения результата вычислений
11
12 SECTION .text
13     GLOBAL _start
14     _start:
15
16     ; Выводим пользователю выражение для вычисления
17     mov eax,msg
18     call sprintf
19
20     ; Выводим запрос на ввод значения с клавиатуры
21     mov eax,request
22     call sprintf
23
24     ; Организуем ввод данных с клавиатуры
25     mov ecx,x
26     mov edx,80
27
28 NORMAL 23 Top 1:1 18:41 calc_function.asm
```

Рис. 3.1: Создание файла calc_function.asm и написание кода для выполнения задания

Проверка программы calc_function на правильность вычислений (рис. 3.2)

```
voffkoz@my-test:~/work/arch-pc/lab06$ nasm -f elf calc_function.asm
voffkoz@my-test:~/work/arch-pc/lab06$ ld -m elf_i386 -o calc_function calc_function.o
voffkoz@my-test:~/work/arch-pc/lab06$ ./calc_function
Вычисление значения функции (8x + 6) * 10
Введите значение x: 1
Результат: 140
voffkoz@my-test:~/work/arch-pc/lab06$ ./calc_function
Вычисление значения функции (8x + 6) * 10
Введите значение x: 4
Результат: 380
```

Рис. 3.2: Результаты запуска программы calc_function и вывод результата работы на экран

3.3 Выводы по результатам выполнения

самостоятельного задания

В ходе выполнения задания для самостоятельной работы была разработана программа для вычисления значения заданной функции $f(x)$ при различных значениях x . При написании программы были применены методы и техники, освоенные в ходе выполнения лабораторной работы, в частности метод обработки ввода пользователя, выводы сообщений на экран и вычисление математических выражений.

3.4 Листинги написанных программ

Листинг программы 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
```

Листинг программы lab6-2.asm

```
%include 'in_out.asm'
```

```
SECTION .text
```

```
GLOBAL _start
```

```
_start:
```

```
mov eax,6
```

```
mov ebx,4
```

```
add eax,ebx
```

```
call iprint
```

```
call quit
```

Листинг программы lab6-3.asm

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
div:  DB 'Результат: ',0
```

```
rem:  DB 'Остаток от деления: ',0
```

```
SECTION .text
```

```
GLOBAL _start
```

```
_start:
```

```

mov  eax,4
mov  ebx,6
mul  ebx
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

```

Листинг программы variant.asm

```

#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 iprintLF
```

```
call quit
```

Листинг программы calc_function.asm

```
%include 'in_out.asm'
```

SECTION .data

```
msg:      DB 'Вычисление значения функции (8x + 6) * 10',0
request:  DB 'Введите значение x: ',0
answer:   DB 'Результат: ',0
```

SECTION .bss

```
x:        RESB 80 ; Значение, которое вводит пользователь
buf1:     RESB 80 ; Промежуточная переменная для хранения результата вычислений
```

SECTION .text

```
GLOBAL _start
```

```
_start:
```

```
; Выводим пользователю выражение для вычисления
```

```
mov eax,msg
```

```
call sprintLF
```

```
; Выводим запрос на ввод значения с клавиатуры
```

```
mov eax,request
```

```
call sprint
```

```
; Организуем ввод данных с клавиатуры
```

```
mov ecx,x
```

```
mov edx,80
```

```
call sread
```

```
; Преобразуем значение, введённое пользователем, в число
```

```
mov eax,x
```

```
call atoi
```

```
; Вычисляем выражение
```

```
mov ebx,8
```

```
mul ebx
```

```
mov ebx,6
```

```
add eax,ebx
```

```
mov ebx,10
```

```
mul ebx
```

```
; Сохраняем результат вычислений в переменную buf1
```

```
mov [buf1],eax
```

```
; Выводим результат вычисления пользователю
```

```
mov eax,answer
```

```
call sprint
```

```
mov eax,[buf1]
```

```
call iprintLF
```

```
call quit
```

4 Выводы

В ходе выполнения лабораторной работы №6 удалось достичь всех поставленных целей, а именно:

- Была изучена работа с численными и символьными данными в языке ассемблера NASM, а также написаны программы для закрепления изученного материала, в частности было продемонстрировано различие в работе с символами и числами, а также использованы разные функции для вывода символов и чисел на экран.
- Была изучена работа с арифметическими операциями в языке ассемблера NASM, были написаны программы для вычисления заданных арифметических выражений, были изучены такие команды как `mul`, `div`, `inc`, `dec`, `neg`, `add`, `sub`.