

Отчет по лабораторной работе №6

дисциплина: Архитектура компьютера

Сергеев Даниил Олегович

Содержание

1	Цель лабораторной работы	4
2	Ход выполнения лабораторной работы	5
3	Ход выполнения заданий для самостоятельной работы	16
4	Вывод	19

Список иллюстраций

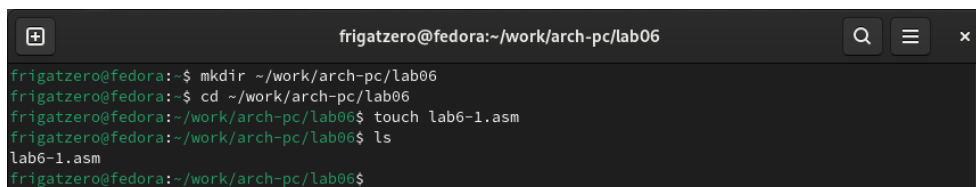
2.1	Создание lab6-1.asm	5
2.2	Код из листинга 6.1	6
2.3	Проверка работы файла lab6-1	6
2.4	Изменяем текст программы lab6-1.asm	7
2.5	Проверка работы измененного файла lab6-1	7
2.6	Создание lab6-2.asm	7
2.7	Код из листинга 6.2	8
2.8	Проверка работы файла lab6-2	8
2.9	Изменяем текст программы lab6-2.asm	9
2.10	Проверка работы измененного файла lab6-2	9
2.11	Изменяем iprintLF на iprint	9
2.12	Проверка работы lab6-2	10
2.13	Создание lab6-3.asm	10
2.14	Код из листинга 6.3	11
2.15	Проверка работы файла lab6-3	11
2.16	Изменяем файл под выражение $f(x) = (4*6 + 2)/5$	12
2.17	Проверка работы измененного файла lab6-3	12
2.18	Создание variant.asm	13
2.19	Код из листинга 6.4	13
2.20	Проверка работы variant.asm	14
3.1	Создаем файл work.asm	16
3.2	Окно редактора Midnight Commander	16
3.3	Проверка работы задания для самостоятельной работы . .	18

1 Цель лабораторной работы

Освоение арифметических инструкций языка ассемблера NASM.

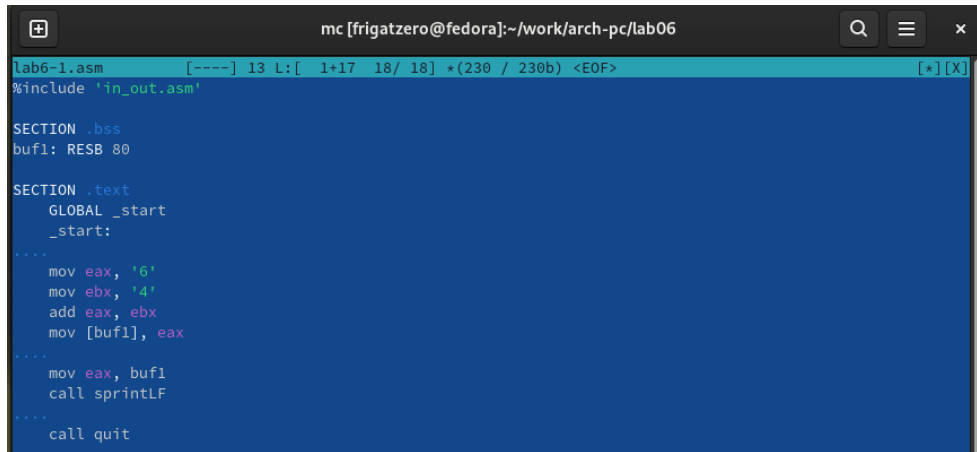
2 Ход выполнения лабораторной работы

1. Создаем каталог `~/work/arch-pc/lab06`, переходим в него и создаем `lab6-1.asm`. В файл вводим текст программы из листинга 6.1.

A screenshot of a terminal window with a dark background. The window title is "frigatzero@fedora:~/work/arch-pc/lab06". The terminal shows a series of commands and their outputs: "mkdir ~/work/arch-pc/lab06", "cd ~/work/arch-pc/lab06", "touch lab6-1.asm", and "ls". The output of "ls" is "lab6-1.asm". The prompt "frigatzero@fedora:~/work/arch-pc/lab06\$" is visible at the bottom.

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

Рис. 2.1: Создание lab6-1.asm

A screenshot of a code editor window titled 'mc [frigatzero@fedora]:~/work/arch-pc/lab06'. The editor shows the assembly file 'lab6-1.asm' with the following code:

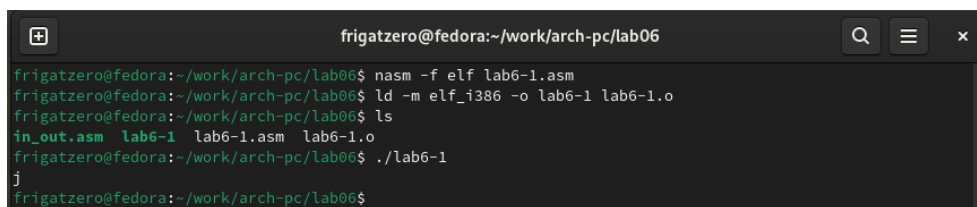
```
lab6-1.asm [----] 13 L: [ 1+17 18/ 18] *(230 / 230b) <EOF> [*][X]
#include 'in_out.asm'

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: Код из листинга 6.1

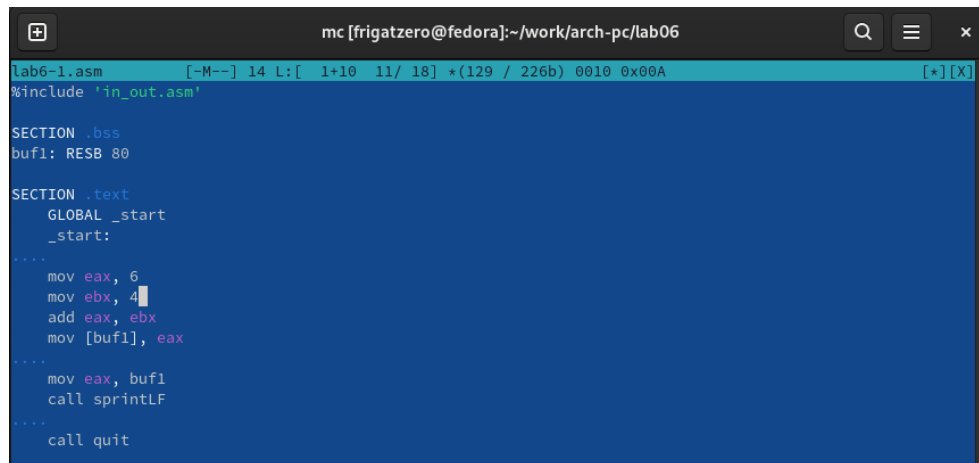
2. Создаем исполняемый файл и запускаем его.

A screenshot of a terminal window titled 'frigatzero@fedora:~/work/arch-pc/lab06'. The terminal shows the following commands and output:

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

Рис. 2.3: Проверка работы файла lab6-1

3. Изменяем текст программы. Запишем в регистры числа вместо символов '4' и '6'. Создаем исполняемый файл и запускаем его. В результате работы программы получаем код символа 10, который соответствует символу переноса строки, поэтому мы не видим его при выводе на экран.

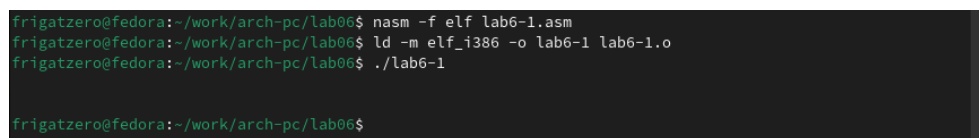


```
lab6-1.asm [-M--] 14 L: [ 1+10 11/ 18] *(129 / 226b) 0010 0x00A [*][X]
#include 'in_out.asm'

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. 2.4: Изменяем текст программы lab6-1.asm

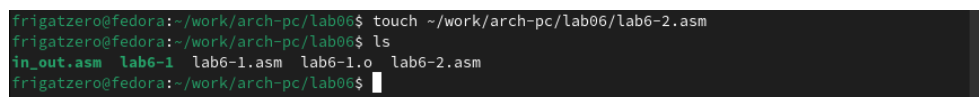


```
frigatzero@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
frigatzero@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
frigatzero@fedora:~/work/arch-pc/lab06$ ./lab6-1

frigatzero@fedora:~/work/arch-pc/lab06$
```

Рис. 2.5: Проверка работы измененного файла lab6-1

4. Создаем файл lab6-2.asm и вводим в него текст из листинга 6.2.



```
frigatzero@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
frigatzero@fedora:~/work/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2.asm
frigatzero@fedora:~/work/arch-pc/lab06$
```

Рис. 2.6: Создание lab6-2.asm

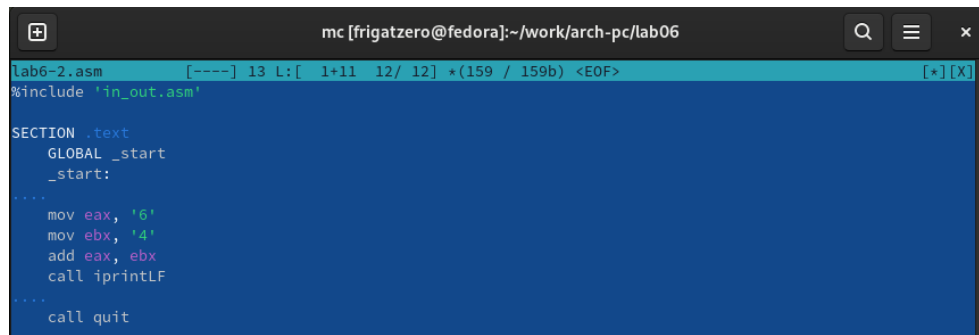
A screenshot of a code editor window titled 'mc [frigatzero@fedora]:~/work/arch-pc/lab06'. The editor shows the contents of 'lab6-2.asm'. At the top, a status bar indicates 'lab6-2.asm' with a line number '13' and a column number '12'. The code includes a header file '%include \'in_out.asm\'', a section definition 'SECTION .text', and a global symbol 'GLOBAL _start'. The main code block starts with '_start:' followed by several instructions: 'mov eax, \'6\'', 'mov ebx, \'4\'', 'add eax, ebx', 'call iprintLF', and 'call quit'.

Рис. 2.7: Код из листинга 6.2

5. Создаем исполняемый файл lab6-2 и запускаем его.

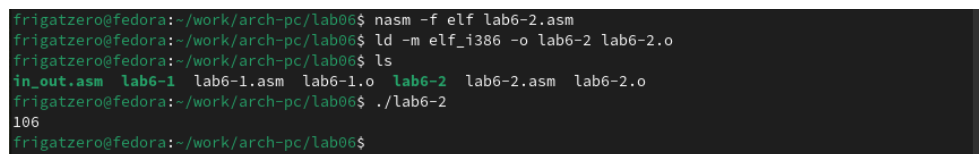
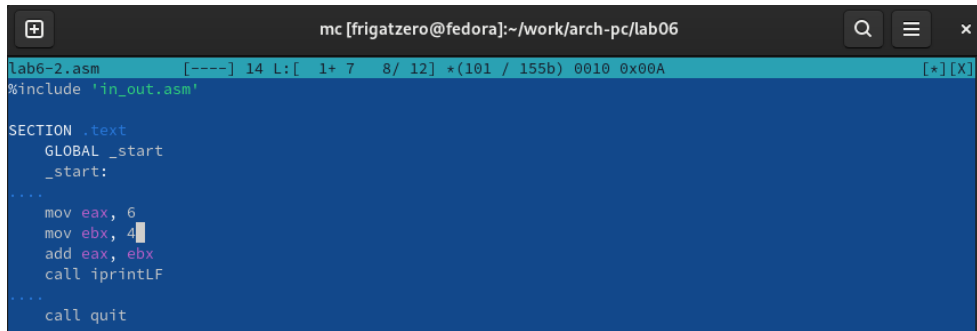
A screenshot of a terminal window showing the compilation and execution of the assembly file. The commands and their outputs are: 'nasm -f elf lab6-2.asm' (no output), 'ld -m elf_i386 -o lab6-2 lab6-2.o' (no output), 'ls' (output: 'in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2 lab6-2.asm lab6-2.o'), './lab6-2' (output: '106'), and the prompt returns to '\$'.

Рис. 2.8: Проверка работы файла lab6-2

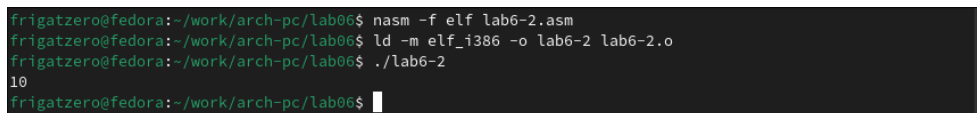
6. Аналогично изменяем символы '4' и '6' на числа. Проверяем работу файла. При исполнении программы получаем результат равный 10.



```
lab6-2.asm [----] 14 L: [ 1+ 7 8/ 12] *(101 / 155b) 0010 0x00A [*][X]
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:
....
mov eax, 6
mov ebx, 4
add eax, ebx
call iprintLF
....
call quit
```

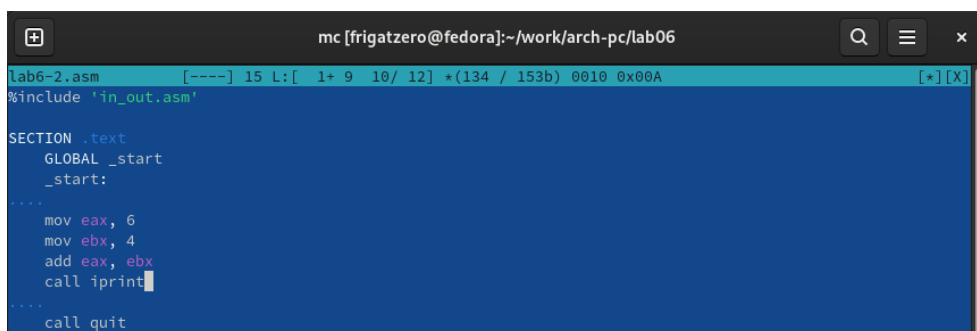
Рис. 2.9: Изменяем текст программы lab6-2.asm



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

Рис. 2.10: Проверка работы измененного файла lab6-2

7. Заменяем функцию iprintLF на iprint. Создаем исполняемый файл и запускаем его. При исполнении программы также получаем 10, но, так как функция iprint не ставит символ переноса строки, сразу после вывода программы начинается строка ввода.



```
lab6-2.asm [----] 15 L: [ 1+ 9 10/ 12] *(134 / 153b) 0010 0x00A [*][X]
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:
....
mov eax, 6
mov ebx, 4
add eax, ebx
call iprint
....
call quit
```

Рис. 2.11: Изменяем iprintLF на iprint

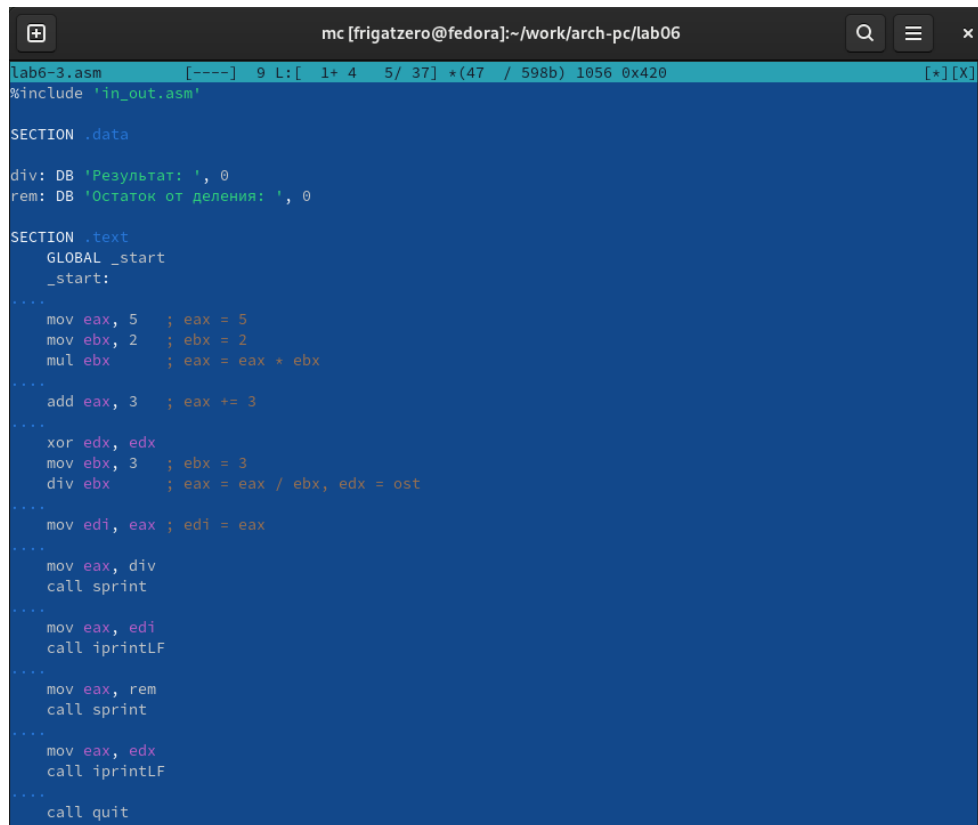
```
frigatzero@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
frigatzero@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
frigatzero@fedora:~/work/arch-pc/lab06$ ./lab6-2
10 frigatzero@fedora:~/work/arch-pc/lab06$
```

Рис. 2.12: Проверка работы lab6-2

8. Создаем файл lab6-3.asm и вводим в него текст из листинга 6.3.

```
frigatzero@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
frigatzero@fedora:~/work/arch-pc/lab06$ ls
in_out.asm lab6-1 lab6-1.asm lab6-1.o lab6-2 lab6-2.asm lab6-2.o lab6-3.asm
frigatzero@fedora:~/work/arch-pc/lab06$
```

Рис. 2.13: Создание lab6-3.asm



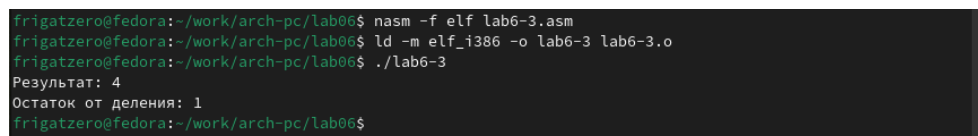
```
lab6-3.asm [----] 9 L: [ 1+ 4 5/ 37] *(47 / 598b) 1056 0x420 [*] [X]
#include 'in_out.asm'

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

SECTION .text
GLOBAL _start
_start:
....
mov eax, 5 ; eax = 5
mov ebx, 2 ; ebx = 2
mul ebx ; eax = eax * ebx
....
add eax, 3 ; eax += 3
....
xor edx, edx
mov ebx, 3 ; ebx = 3
div ebx ; eax = eax / ebx, edx = ost
....
mov edi, eax ; 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: Код из листинга 6.3

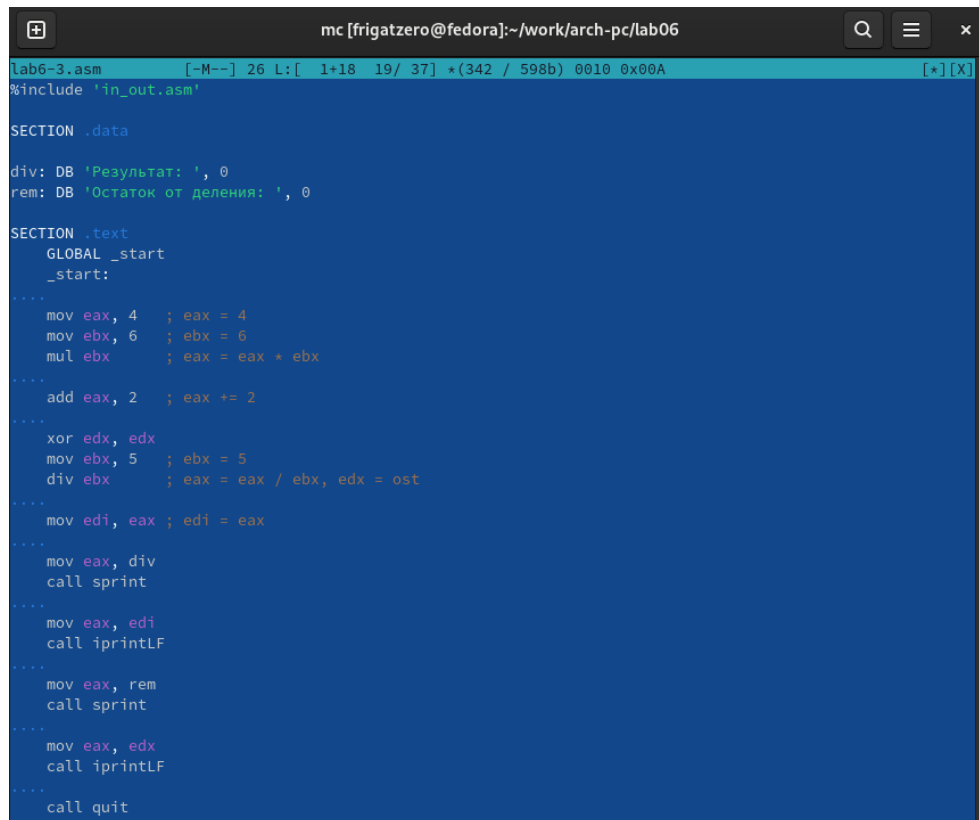
9. Создаем исполняемый файл lab6-3 и запускаем его.



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

Рис. 2.15: Проверка работы файла lab6-3

10. Изменяем текст программы для вычисления выражения $f(x) = (4 \cdot 6 + 2) / 5$. Создаем исполняемый файл и проверяем его работу.

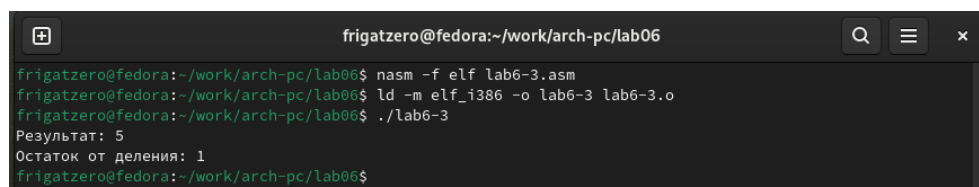


```
lab6-3.asm [-M--] 26 L: [ 1+18 19/ 37] *(342 / 598b) 0010 0x00A [*][X]
#include 'in_out.asm'

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

SECTION .text
GLOBAL _start
_start:
....
mov eax, 4 ; eax = 4
mov ebx, 6 ; ebx = 6
mul ebx ; eax = eax * ebx
....
add eax, 2 ; eax += 2
....
xor edx, edx
mov ebx, 5 ; ebx = 5
div ebx ; eax = eax / ebx, edx = ost
....
mov edi, eax ; 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: Изменяем файл под выражение $f(x) = (4*6 + 2)/5$



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

Рис. 2.17: Проверка работы измененного файла lab6-3

11. Создаем файл variant.asm и вводим в него текст из листинга 6.4.

```
frigatzero@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
frigatzero@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm  lab6-2      lab6-2.o    lab6-3.asm  variant.asm
lab6-1      lab6-1.o    lab6-2.asm  lab6-3      lab6-3.o
frigatzero@fedora:~/work/arch-pc/lab06$
```

Рис. 2.18: Создание variant.asm

```
variant.asm [----] 13 L: [ 1+34 35/ 35] *(406 / 406b) <EOF> [*][X]
#include 'in_out.asm'

SECTION .data
msg: DB
rem: DB

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

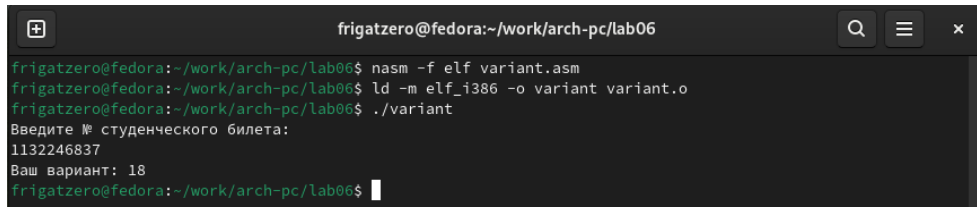
mov eax, rem
call sprint

mov eax, edx
call iprintf

call quit
```

Рис. 2.19: Код из листинга 6.4

12. Создаем исполняемый файл, проверяем его. Вычисляем номер варианта задания для самостоятельной работы. Результат работы программы и аналитическое решение совпали.



```
frigatzero@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
frigatzero@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
frigatzero@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246837
Ваш вариант: 18
frigatzero@fedora:~/work/arch-pc/lab06$
```

Рис. 2.20: Проверка работы variant.asm

13. Ответы на вопросы:

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

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

- 2. Данные инструкции используются для записи введенной строки в переменную x.
- 3. Инструкция call atoi используется для вызова функции, которая преобразовывает ascii-код символа в целое число.
- 4. За вычисление варианта отвечают 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 увеличения значения указанного регистра на единицу.

- 7. За вывод на экран результата вычислений отвечают строки:

```
mov eax, edx  
call iprintLF
```

3 Ход выполнения заданий для самостоятельной работы

1. Создаем программу для вычисления выражения №18 из таблицы 6.3. Для этого создаем файл `work.asm` и в окне `mc` пишем код программы

```
frigatzero@fedora:~/work/arch-pc/lab06
frigatzero@fedora:~/work/arch-pc/lab06$ touch work.asm
frigatzero@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm  lab6-2.asm  lab6-2.o  lab6-3.asm  variant.o  variant.o
lab6-1      lab6-1.o   lab6-2.asm  lab6-3   lab6-3.o    variant.asm  work.asm
frigatzero@fedora:~/work/arch-pc/lab06$
```

Рис. 3.1: Создаем файл `work.asm`

mc [frigatzero@fedora]:~/work/arch-pc/lab06

work.asm [----] 0 L:[1+ 0 1/ 1] *(0 / 0b) <EOF> [*] [X]

1 Help 2 Save 3 Mark 4 Replac 5 Conv 6 Move 7 Search 8 Delete 9 PullDn 10 Quit

Рис. 3.2: Окно редактора Midnight Commander

Листинг 4.1 Программа для вычисления выражения $y = f(x)$.

```
%include 'in_out.asm'

SECTION .data
exp: DB 'Программа выводит значение выражения №18  $f(x) = 3(x+10)-20$ . ',0
inp: DB 'Введите x: ',0
ans: DB 'Результат вычислений: ',0

SECTION .bss
x: RESB 80

SECTION .text
    GLOBAL _start
    _start:

    mov eax, exp ; Вывод сообщения exp
    call printf ;

    mov eax, inp ; Вывод сообщения inp
    call printf ;

    mov ecx, x ; Записываем символы
    mov edx, 80 ; из ввода в переменную
    call read ; (буфер) x

    mov eax, x ; Переводим значение x из
```

```

call atoi      ; кода ascii в число и
               ; и записываем его в eax

add eax, 10    ; x + 10

mov ebx, 3     ;
mul ebx        ; 3 * (x + 10)

sub eax, 20    ; 3 * (x + 10) - 20

mov edi, eax   ; Сохраняем результат в edi

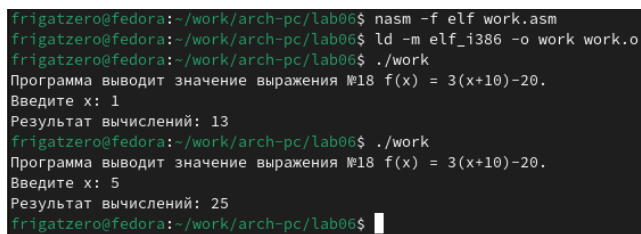
mov eax, ans   ; Вывод сообщения ans
call sprint    ;

mov eax, edi   ; Вывод результата сразу
call iprintLF  ; после ans

call quit

```

2. Проверяем работу программы для x равного 1 и 5.



```

frigatzero@fedora:~/work/arch-pc/lab06$ nasm -f elf work.asm
frigatzero@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o work work.o
frigatzero@fedora:~/work/arch-pc/lab06$ ./work
Программа выводит значение выражения №18 f(x) = 3(x+10)-20.
Введите x: 1
Результат вычислений: 13
frigatzero@fedora:~/work/arch-pc/lab06$ ./work
Программа выводит значение выражения №18 f(x) = 3(x+10)-20.
Введите x: 5
Результат вычислений: 25
frigatzero@fedora:~/work/arch-pc/lab06$

```

Рис. 3.3: Проверка работы задания для самостоятельной работы

4 Вывод

После выполнения заданий лабораторной работы и заданий для самостоятельной работы я освоил арифметические инструкции `add`, `sub`, `mul`, `div`, `inc` и `dec`, приобрел опыт в написании собственных программ для вычисления выражений на языке ассемблера NASM.