

Лабораторная работа №6

Архитектура компьютера

Мурашов Иван Вячеславович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Символьные и численные данные в NASM	7
3.2	Выполнение арифметических операций в NASM	12
3.2.1	Ответы на вопросы по программе	15
3.3	Выполнение заданий для самостоятельной работы	17
4	Выводы	21

Список иллюстраций

3.1	Создание каталога и файла в нём	7
3.2	Окно Midnight Commander. Редактор mcedit	8
3.3	Трансляция, компоновка и запуск файлов	8
3.4	Окно Midnight Commander. Редактор mcedit	9
3.5	Трансляция, компоновка и запуск файлов	9
3.6	Создание файла	9
3.7	Окно Midnight Commander. Редактор mcedit	10
3.8	Трансляция, компоновка и запуск файлов	10
3.9	Окно Midnight Commander. Редактор mcedit	11
3.10	Трансляция, компоновка и запуск файлов	11
3.11	Окно Midnight Commander. Редактор mcedit	12
3.12	Трансляция, компоновка и запуск файлов	12
3.13	Создание файла	12
3.14	Окно Midnight Commander. Редактор mcedit	13
3.15	Трансляция, компоновка и запуск файлов	13
3.16	Окно Midnight Commander. Редактор mcedit	14
3.17	Трансляция, компоновка и запуск файлов	14
3.18	Создание файла	14
3.19	Окно Midnight Commander. Редактор mcedit	15
3.20	Трансляция, компоновка и запуск файлов	15
3.21	Окно Midnight Commander. Редактор mcedit	16
3.22	Окно Midnight Commander. Редактор mcedit	16
3.23	Окно Midnight Commander. Редактор mcedit	17
3.24	Создание файла	17
3.25	Окно Midnight Commander. Редактор mcedit	18
3.26	Трансляция, компоновка и запуск файлов	19

Список таблиц

1 Цель работы

Целью данной лабораторной работы является освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

3 Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

Создаю каталог для программ лабораторной работы №6, перехожу в него и создаю файл lab6-1.asm (рис. [3.1]).

```
[ivmurashov@fedora ~]$ mkdir ~/work/arch-pc/lab06  
[ivmurashov@fedora ~]$ cd ~/work/arch-pc/lab06  
[ivmurashov@fedora lab06]$ touch lab6-1.asm
```

Рис. 3.1: Создание каталога и файла в нём

Открываю файл lab6-1.asm для редактирования в mcedit. Ввожу в файл текст программы из листинга 6.1 (рис. [3.2]).

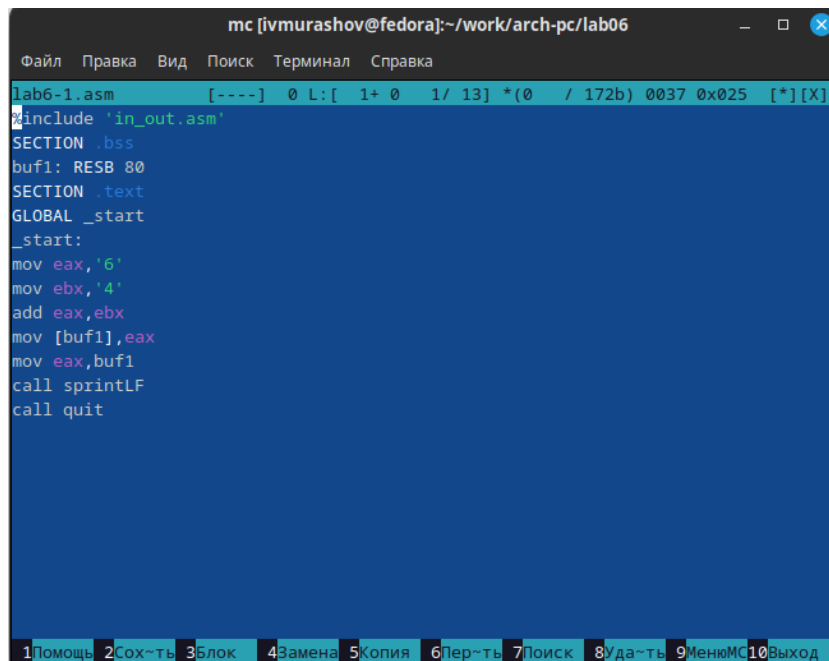


Рис. 3.2: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. Вывод программы: символ j, так как программа вывела символ, соответствующий сумме двоичных кодов символов 4 и 6 по системе ASCII (рис. [3.3]).

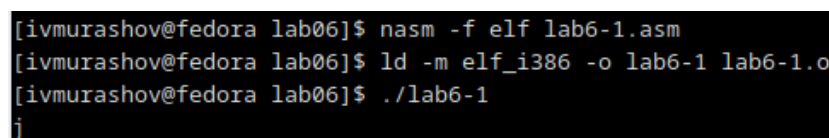


Рис. 3.3: Трансляция, компоновка и запуск файлов

Изменяю текст программы, записав в регистры числа вместо символов. (рис. [3.4]).

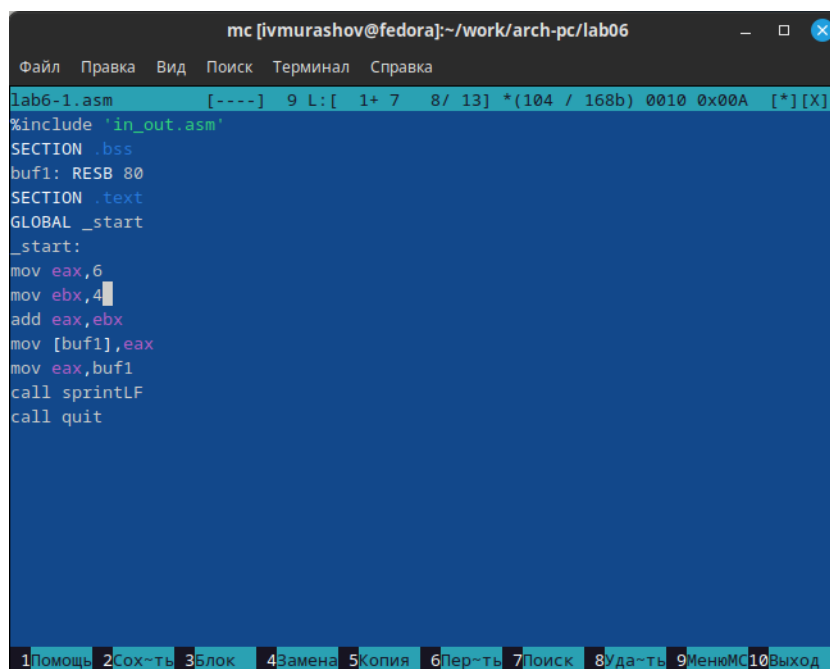


Рис. 3.4: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. В данном случае выводится символ с кодом 10, которое соответствует символу “LF,\n” по таблице ASCII. Символ не отображается при выводе на экран, так как является символом перевода строки (рис. [3.5]).

```
[ivmurashov@fedora lab06]$ nasm -f elf lab6-1.asm
[ivmurashov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[ivmurashov@fedora lab06]$ ./lab6-1
```

Рис. 3.5: Трансляция, компоновка и запуск файлов

Создаю файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. [3.6]).

```
[ivmurashov@fedora lab06]$ touch lab6-2.asm
```

Рис. 3.6: Создание файла

Ввожу в данный файл текст программы из листинга 6.2 (рис. [3.7]).

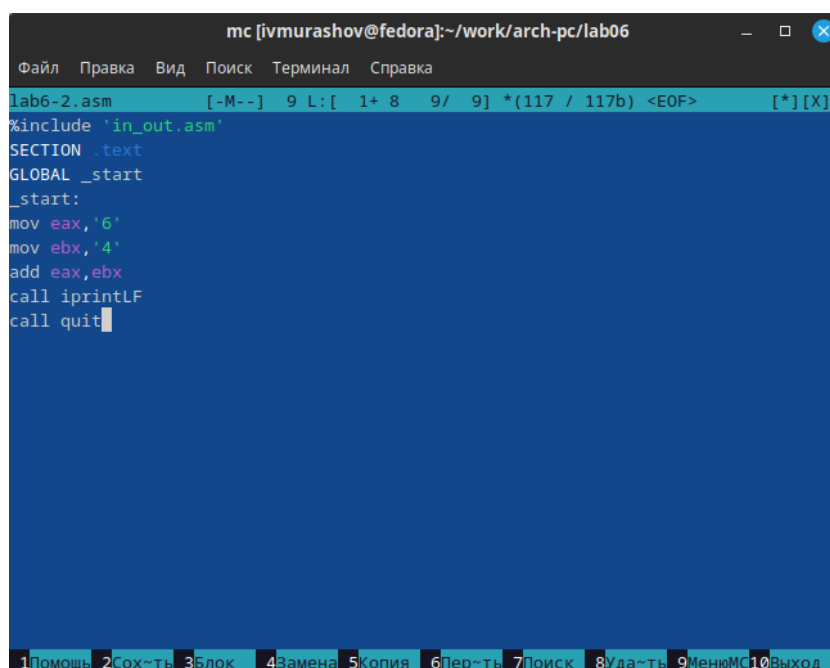


Рис. 3.7: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. В результате работы программы я получаю число 106. В отличие от программы из листинга 6.1, функция `iprintLF` позволяет вывести число, а не символ, кодом которого является это число (рис. [3.8]).

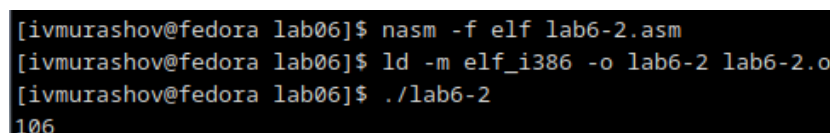


Рис. 3.8: Трансляция, компоновка и запуск файлов

Изменяю текст программы, записав в регистры числа вместо символов. (рис. [3.9]).

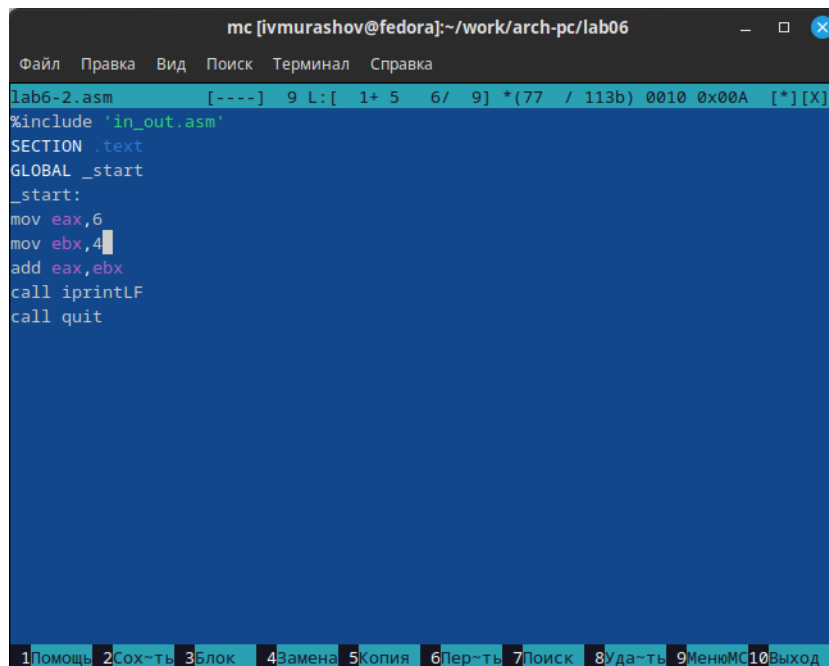


Рис. 3.9: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. В результате выводится число 10, так как программа складывает не коды в системе ASCII, а сами числа (рис. [3.10]).

```

[ivmurashov@fedora lab06]$ nasm -f elf lab6-2.asm
[ivmurashov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[ivmurashov@fedora lab06]$ ./lab6-2
10

```

Рис. 3.10: Трансляция, компоновка и запуск файлов

Заменяю функцию iprintLF на iprint (рис. [3.11]).

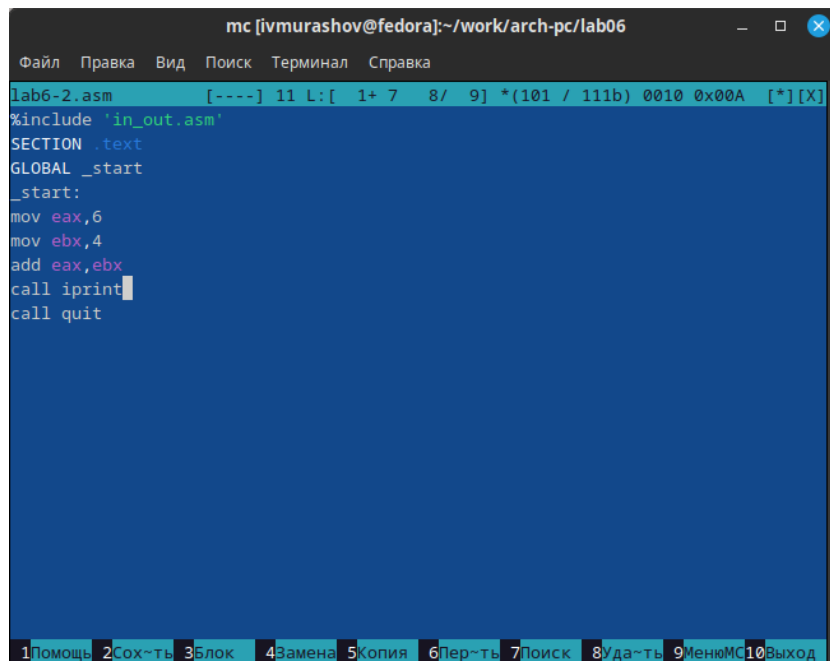


Рис. 3.11: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. Вывод функций `iprintLF` и `iprint` отличается тем, что в последнем случае не происходит перехода на новую строку при выводе значения (рис. [3.12]).

```
[ivmurashov@fedora lab06]$ nasm -f elf lab6-2.asm
[ivmurashov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[ivmurashov@fedora lab06]$ ./lab6-2
10[ivmurashov@fedora lab06]$
```

Рис. 3.12: Трансляция, компоновка и запуск файлов

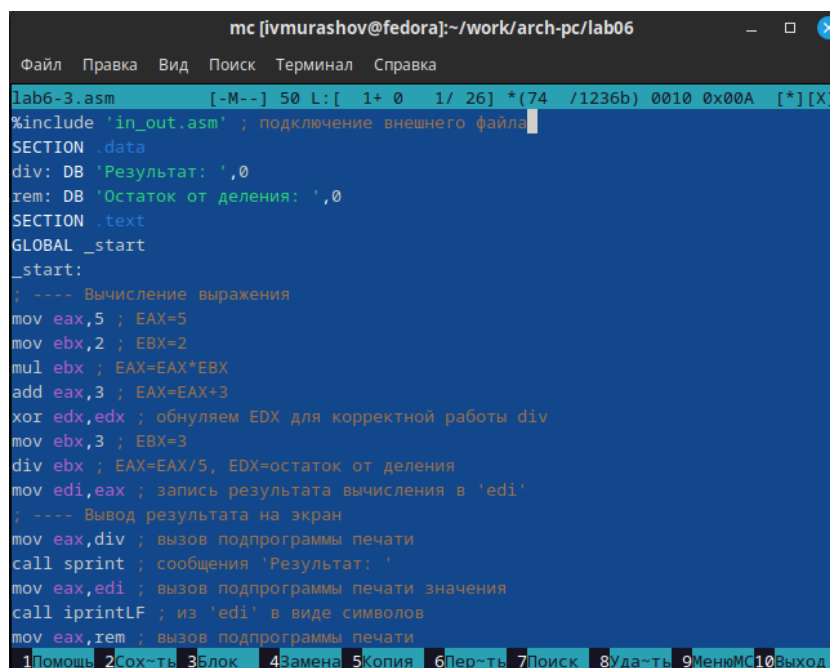
3.2 Выполнение арифметических операций в NASM

Создаю файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06` (рис. [3.13]).

```
[ivmurashov@fedora ~]$ touch ~/work/arch-pc/lab06/lab6-3.asm
```

Рис. 3.13: Создание файла

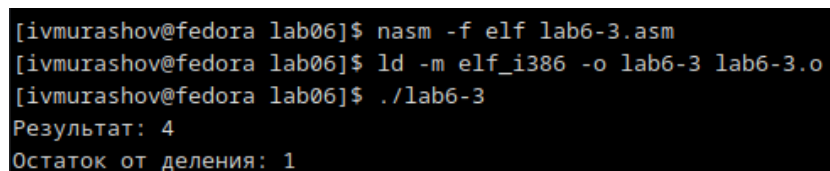
Ввожу в файл текст программы из листинга 6.3 (рис. [3.14]).



```
mc [ivmurashov@fedora]:~/work/arch-pc/lab06
Файл  Правка  Вид  Поиск  Терминал  Справка
lab6-3.asm  [-M--] 50 L:[ 1+ 0 1/ 26] *(74 /1236b) 0010 0x00A [*][X]
#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/5, 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
1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход
```

Рис. 3.14: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его (рис. [3.15]).



```
[ivmurashov@fedora lab06]$ nasm -f elf lab6-3.asm
[ivmurashov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[ivmurashov@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 3.15: Трансляция, компоновка и запуск файлов

Изменяю текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$ (рис. [3.16]).

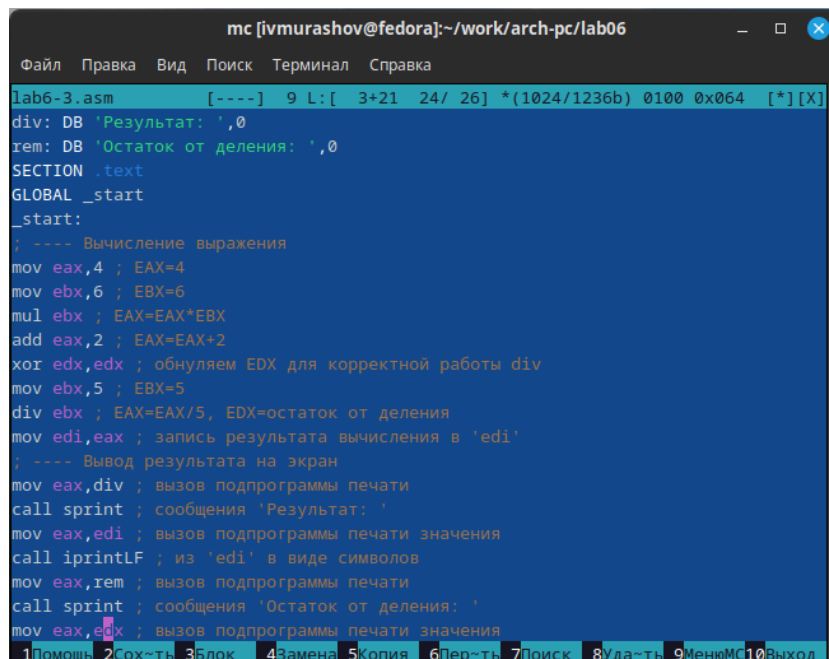


Рис. 3.16: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. При самостоятельном расчёте результат совпал с ранее полученным выводом (рис. [3.17]).

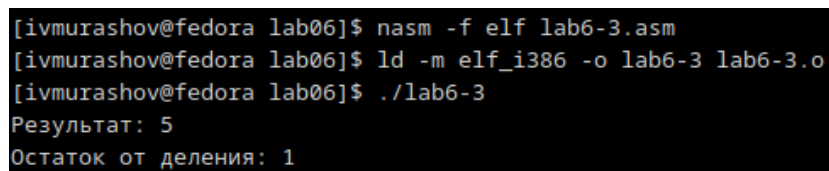


Рис. 3.17: Трансляция, компоновка и запуск файлов

Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. [3.18]).

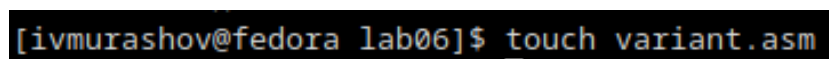


Рис. 3.18: Создание файла

Ввожу в файл текст программы из листинга 6.4 (рис. [3.19]).

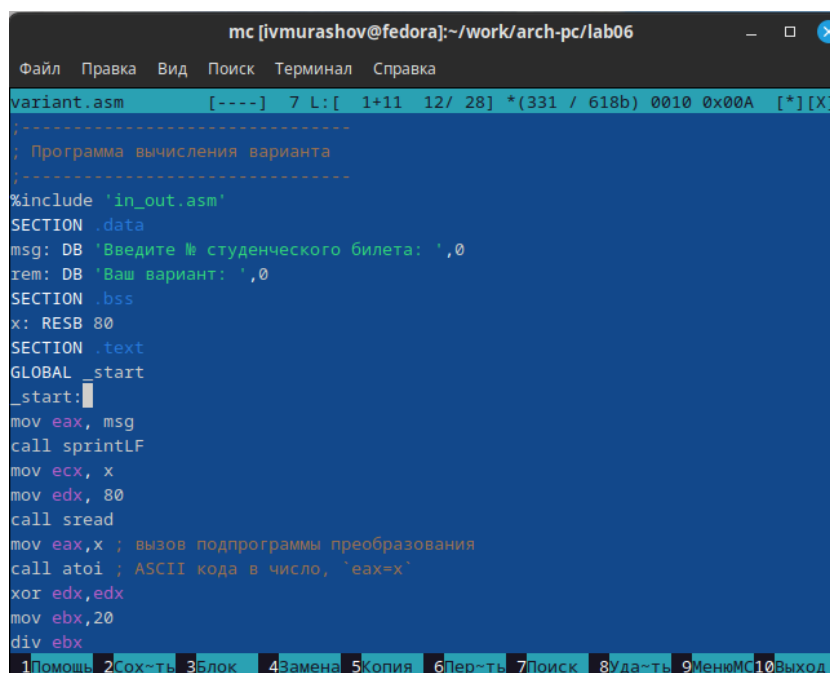


Рис. 3.19: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. Проверяю результат работы программы, вычислив номер своего варианта аналитически. Результаты совпали (рис. [3.20]).

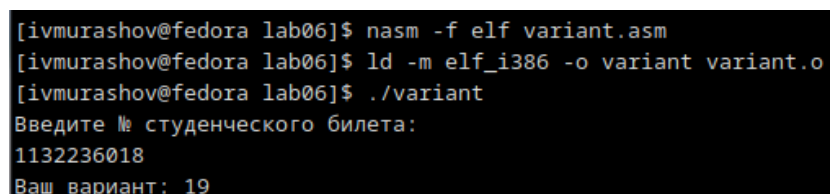
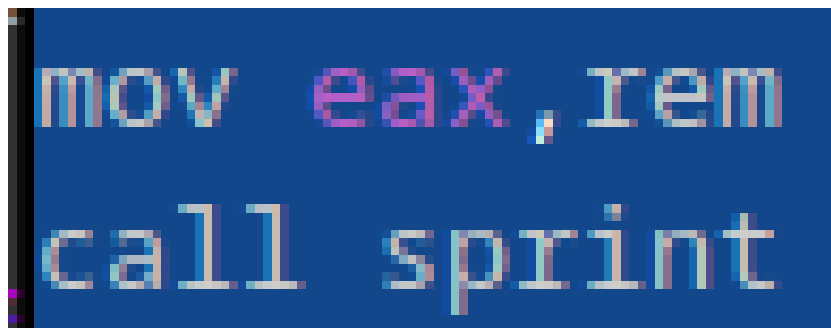


Рис. 3.20: Трансляция, компоновка и запуск файлов

3.2.1 Ответы на вопросы по программе

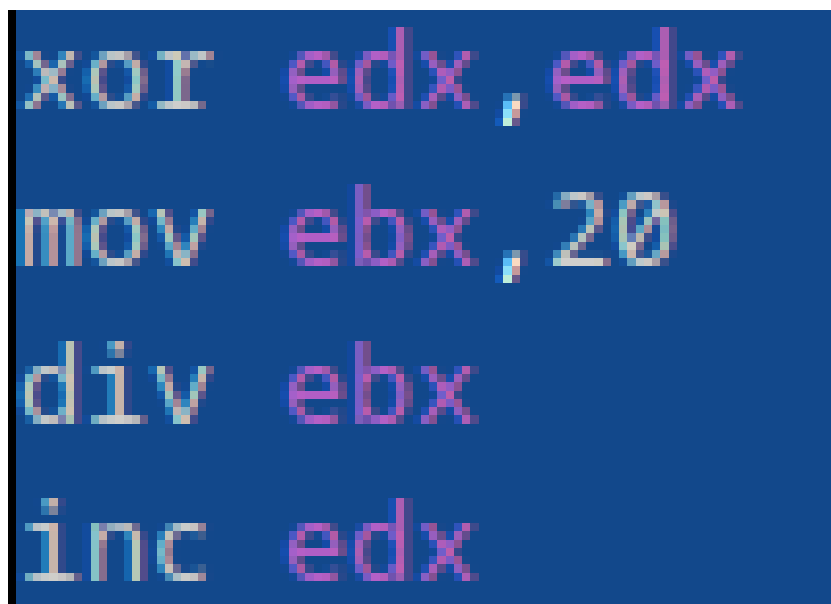
1. За вывод на экран сообщения 'Ваш вариант:' отвечают следующие строки:

A screenshot of the Midnight Commander file manager's built-in editor, mcedit. The background is dark blue. The text is in a monospaced font with a rainbow color scheme: 'mov' is cyan, 'eax' is magenta, 'rem' is cyan, 'call' is cyan, and 'sprint' is magenta. The code consists of two lines: 'mov eax,rem' and 'call sprint'.

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

Рис. 3.21: Окно Midnight Commander. Редактор mcedit

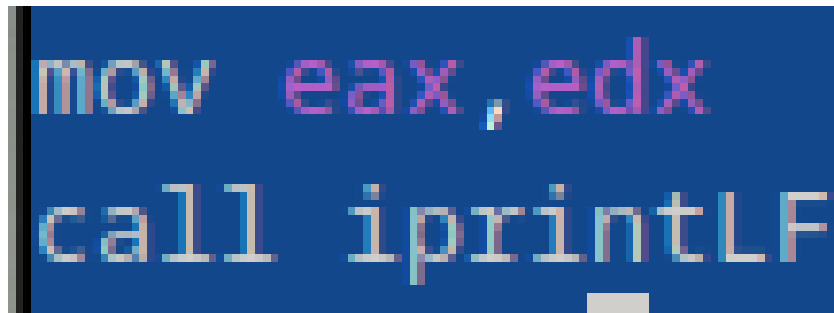
2. Данные инструкции используются для того, чтобы поместить адрес вводимой строки в регистр `ecx` и записать в регистр `edx` длину вводимой строки (80 bytes). Затем мы вызываем подпрограмму из внешнего файла `in_out.asm` для ввода значения `x` с клавиатуры.
3. Инструкция `'call atoi'` используется для перевода ASCII-код символа в целое число.
4. Зы вычисление варианта отвечают следующие строки листинга 6.4:

A screenshot of the Midnight Commander file manager's built-in editor, mcedit. The background is dark blue. The text is in a monospaced font with a rainbow color scheme: 'xor' is cyan, 'edx' is magenta, 'mov' is cyan, 'ebx' is magenta, 'div' is cyan, 'ebx' is magenta, 'inc' is cyan, and 'edx' is magenta. The code consists of four lines: 'xor edx,edx', 'mov ebx,20', 'div ebx', and 'inc edx'.

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

Рис. 3.22: Окно Midnight Commander. Редактор mcedit

5. Остаток от деления при выполнении инструкции 'div ebx' записывается в регистр edx.
6. Инструкция 'inc edx' используется для прибавления 1 к значению, записанному в регистре edx.
7. За вывод на экран результата вычислений отвечают следующие строки листинга 6.4:

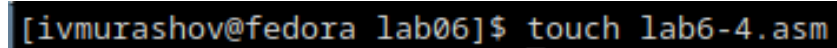


```
mov eax,edx
call iprintLF
```

Рис. 3.23: Окно Midnight Commander. Редактор mcedit

3.3 Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm в каталоге ~/work/arch-pc/lab06 (рис. [3.24]).



```
[ivmurashov@fedora lab06]$ touch lab6-4.asm
```

Рис. 3.24: Создание файла

Открываю созданный файл с помощью редактора tc для редактирования и ввожу в него текст программы для вычисления выражения $((\frac{1}{3}x + 5) \cdot 7) \cdot 19$ (вариант) (рис. [3.25]).

```
mc [ivmurashov@fedora]:~/work/arch-pc/lab06
Файл  Правка  Вид  Поиск  Терминал  Справка
lab6-4.asm  [----] 27 L:[ 1+38 39/ 41] *(1147/1149b) 0010 0x00A [*][X]
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
    msg_str:
        DB 'Введите значение переменной: ',0
    res_str:
        DB 'Результат: ',0
SECTION .bss
    x:
        RESB 16 ; Задаём переменную x
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
    mov eax, msg_str ; string addr
    call sprintf      ; print string

    mov edx, 15      ; buffer size
    mov ecx, x        ; buffer addr
    call sread       ; read string from stdin to buffer

    mov eax, x        ; string addr
    call atoi         ; convert to integer
; ----
    xor edx, edx      ; move edx, 0 (обнуление)
    mov ebx, 3
    div ebx           ; div eax, ebx ( by 3 )

    add eax, 5        ; plus 5

    mov ebx, 7
    mul ebx           ; mul eax, ebx (7)
    mov edi, eax      ; save eax to edi
; ---- Вывод результата на экран
    mov eax, res_str ;
    call sprintf      ; print 'Результат: '
    mov eax, edi      ; restore eax
    call iprintLF     ; print eax as string
    call quit         ; exit
```

Рис. 3.25: Окно Midnight Commander. Редактор mcedit

Создаю исполняемый файл и запускаю его. Проверяю корректность работы программы, вычислив получаемые значения вручную. Результаты совпали (рис. [3.26]).

```

[ivmurashov@fedora lab06]$ ./lab6-4
Введите значение переменной:
3
Результат: 42
[ivmurashov@fedora lab06]$ ./lab6-4
Введите значение переменной:
9
Результат: 56

```

Рис. 3.26: Трансляция, компоновка и запуск файлов

Листинг 4. Программа для вычисления значения выражения $((1/3)x + 5)$
7)

```

#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
    msg_str:
        DB 'Введите значение переменной: ',0
    res_str:
        DB 'Результат: ',0
SECTION .bss
    x:
        RESB 16 ; Задаём переменную x
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
    mov eax, msg_str ; string addr
    call sprintf      ; print string

```

```

mov edx, 15      ; buffer size
mov ecx, x       ; buffer addr
call sread       ; read string from stdin to buffer

mov eax,x        ; string addr
call atoi        ; convert to integer

xor edx, edx     ; move edx, 0 (обнуление)
mov ebx, 3
div ebx          ; div eax, ebx ( by 3 )

add eax, 5       ; plus 5

mov ebx, 7       ;
mul ebx          ; mul eax, ebx (7)
mov edi, eax     ; save eax to edi

; ---- Вывод результата на экран
mov eax, res_str ;
call sprintf     ; print 'Результат: '
mov eax, edi     ; restore eax
call iprintLF    ; print eax as string
call quit       ; exit

```

4 Выводы

В ходе выполнения данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.