

Отчёт по лабораторной работе №6

Ярослав Антонович Меркулов

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Символьные и численные данные в NASM	5
2.2	Выполнение арифметических операций в NASM	11
2.3	Ответы на вопросы	15
3	Выполнение самостоятельной работы	16
4	Выводы	18

Список иллюстраций

2.1	Создание каталога, переход в него, создание файла	5
2.2	Введённый текст программы	6
2.3	Работа программы	6
2.4	Изменённый файл	7
2.5	Работа новой программы	7
2.6	Файл lab6-2.asm	8
2.7	Работа программы	8
2.8	Изменённый lab6-2.asm	9
2.9	Работа программы	9
2.10	Изменённый lab6-2.asm	10
2.11	Работа программы	10
2.12	Создание файла	11
2.13	Готовый файл lab6-3.asm	12
2.14	Работа программы	12
2.15	Изменённый файл lab6-3.asm	13
2.16	Работа программы	13
2.17	Файл variant.asm	14
2.18	Работа программы	14
3.1	Готовый fx.asm	16
3.2	Работа программы	17

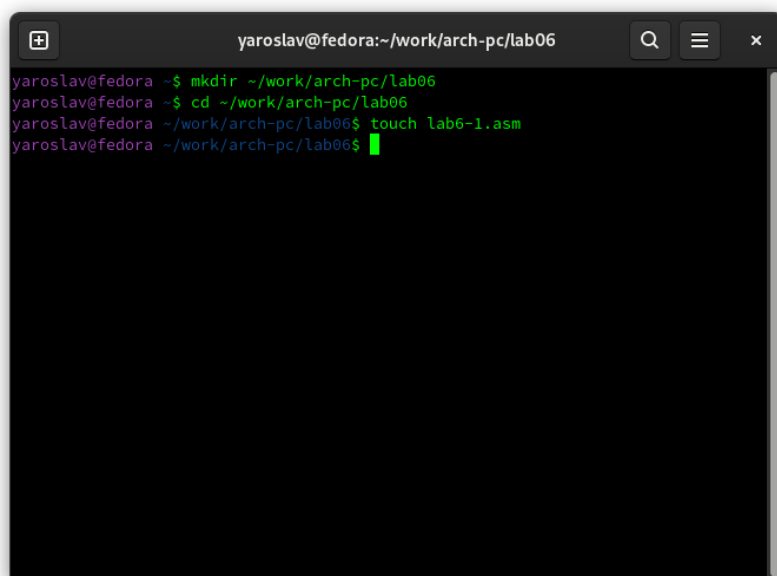
1 Цель работы

Освоить арифметические инструкции языка программирования NASM

2 Выполнение лабораторной работы

2.1 Символьные и численные данные в NASM

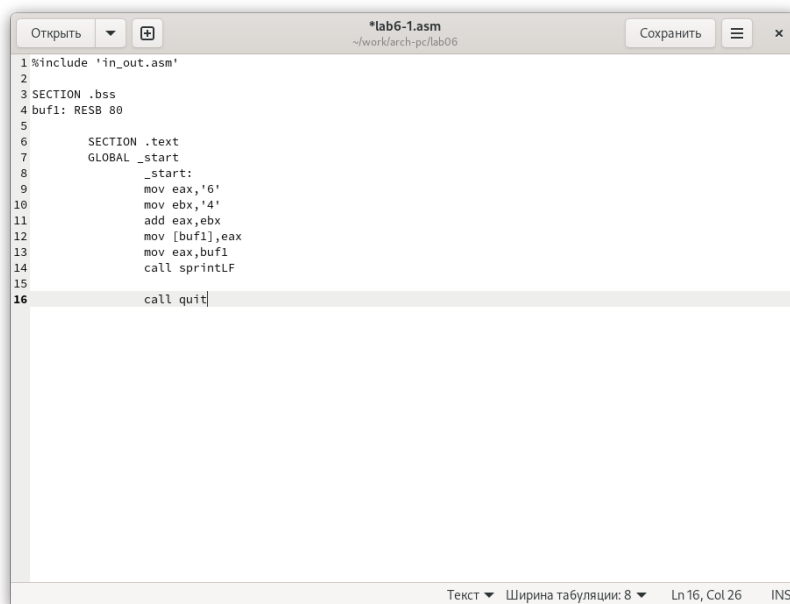
1. Создаём каталог для лабораторной работы, переходим в него и создаём файл lab6-1.asm.



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

Рис. 2.1: Создание каталога, переход в него, создание файла

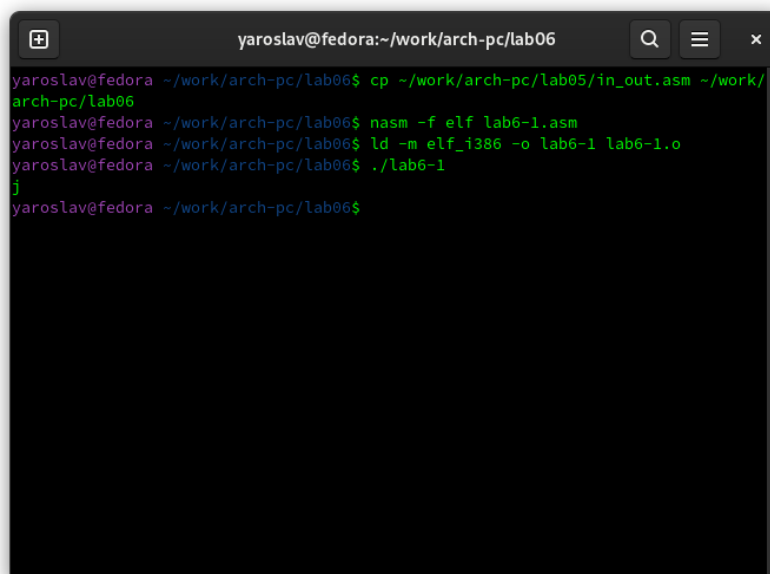
2. Вводим текст программы из листинга 6.1.



```
1 %include 'in_out.asm'
2
3 SECTION .bss
4 buf1: RESB 80
5
6 SECTION .text
7 GLOBAL _start
8 _start:
9     mov eax, '6'
10    mov ebx, '4'
11    add eax, ebx
12    mov [buf1], eax
13    mov eax, buf1
14    call sprintf
15
16    call quit
```

Рис. 2.2: Введённый текст программы

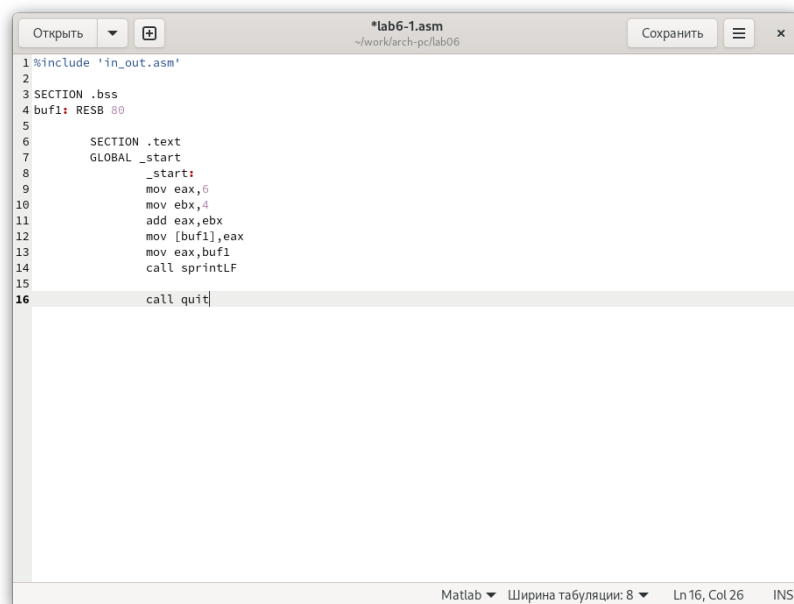
3. Создаём исполняемый файл и запускаем. В результате выводится символ j.



```
yaroslav@fedora: ~/work/arch-pc/lab06
yaroslav@fedora ~/work/arch-pc/lab06$ cp ~/work/arch-pc/lab05/in_out.asm ~/work/arch-pc/lab06
yaroslav@fedora ~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
yaroslav@fedora ~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
yaroslav@fedora ~/work/arch-pc/lab06$ ./lab6-1
j
yaroslav@fedora ~/work/arch-pc/lab06$
```

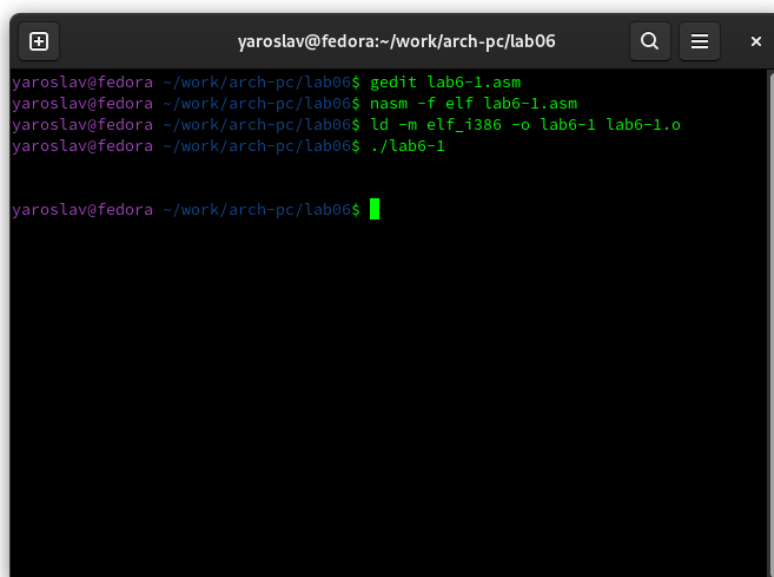
Рис. 2.3: Работа программы

4. Изменяем текст программы (убираем одинарные кавычки), создаём исполняемый файл и запускаем. Вывелся символ переноса строки.



```
1 %include "in_out.asm"
2
3 SECTION .bss
4 buf1: RESB 80
5
6 SECTION .text
7 GLOBAL _start
8 _start:
9     mov eax, 6
10    mov ebx, 4
11    add eax, ebx
12    mov [buf1], eax
13    mov eax, buf1
14    call printf
15
16    call _exit
```

Рис. 2.4: Изменённый файл

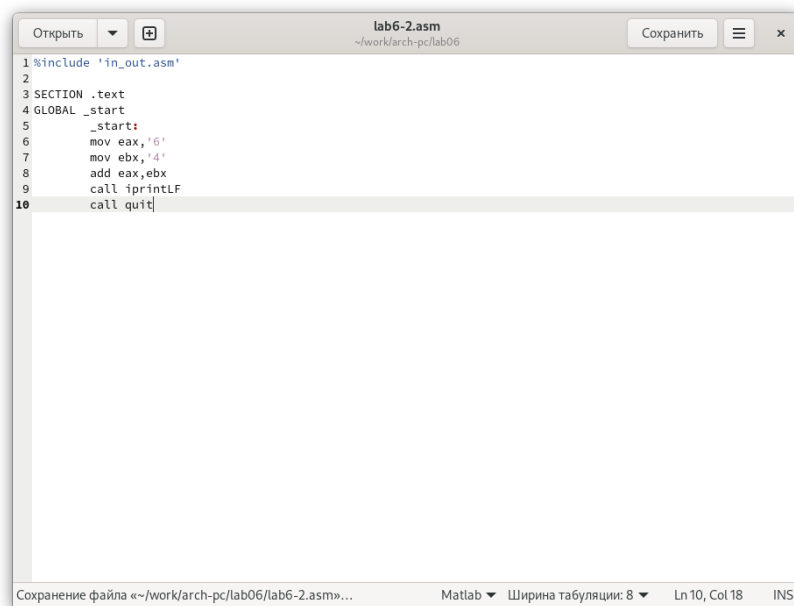


```
yaroslav@fedora: ~/work/arch-pc/lab06
yaroslav@fedora ~/work/arch-pc/lab06$ gedit lab6-1.asm
yaroslav@fedora ~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
yaroslav@fedora ~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
yaroslav@fedora ~/work/arch-pc/lab06$ ./lab6-1

yaroslav@fedora ~/work/arch-pc/lab06$
```

Рис. 2.5: Работа новой программы

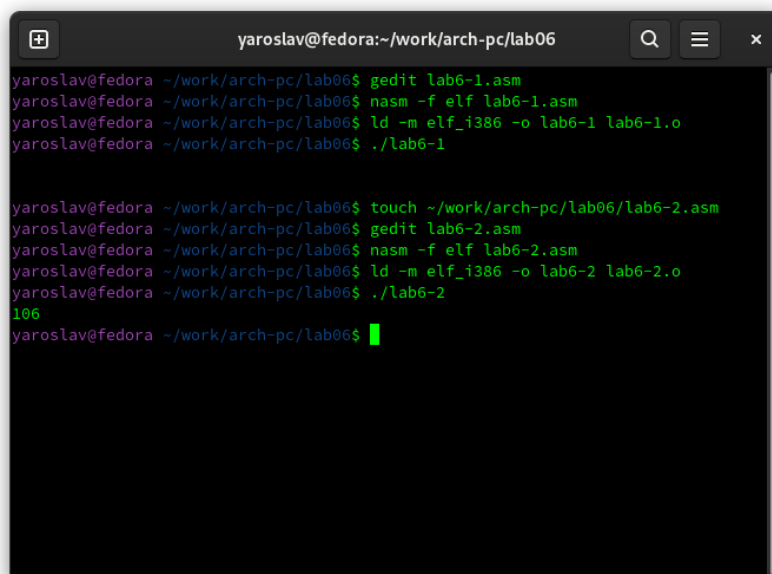
5. Создаём новый файл и переносим туда текст из листинга 6.2.



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

Рис. 2.6: Файл lab6-2.asm

6. Создаём исполняемый файл и запускаем. Видим число 106.

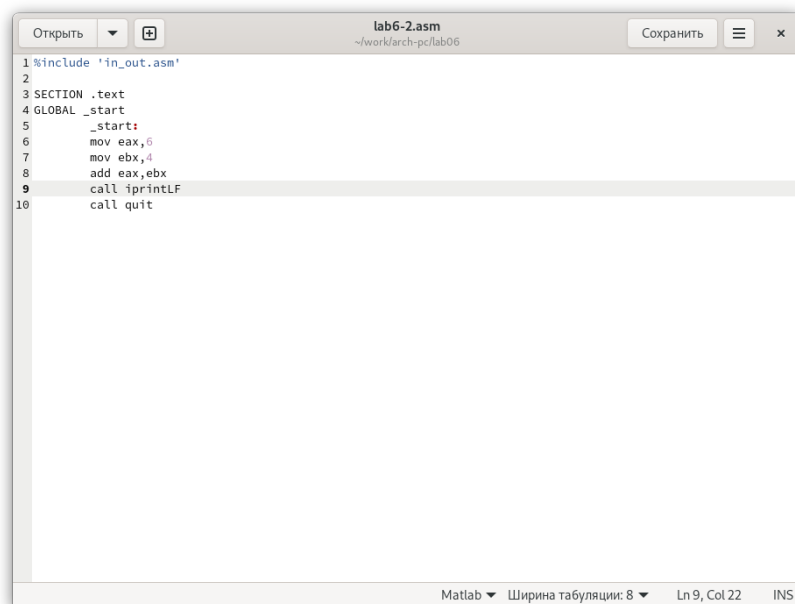


```
yaroslav@fedora: ~/work/arch-pc/lab06
yaroslav@fedora: ~/work/arch-pc/lab06$ gedit lab6-1.asm
yaroslav@fedora: ~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
yaroslav@fedora: ~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
yaroslav@fedora: ~/work/arch-pc/lab06$ ./lab6-1

yaroslav@fedora: ~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
yaroslav@fedora: ~/work/arch-pc/lab06$ gedit lab6-2.asm
yaroslav@fedora: ~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
yaroslav@fedora: ~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
yaroslav@fedora: ~/work/arch-pc/lab06$ ./lab6-2
106
yaroslav@fedora: ~/work/arch-pc/lab06$
```

Рис. 2.7: Работа программы

7. Удаляем одинарные кавычки (как в пункте 4).

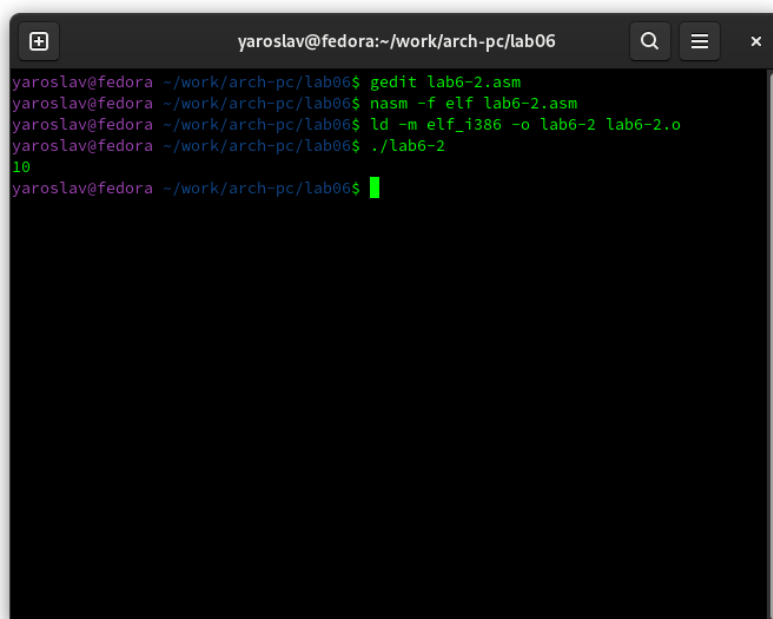


The screenshot shows a text editor window titled "lab6-2.asm" with the file path "~/work/arch-pc/lab06". The code is as follows:

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

The status bar at the bottom indicates "Matlab", "Ширина табуляции: 8", "Ln 9, Col 22", and "INS".

Рис. 2.8: Изменённый lab6-2.asm

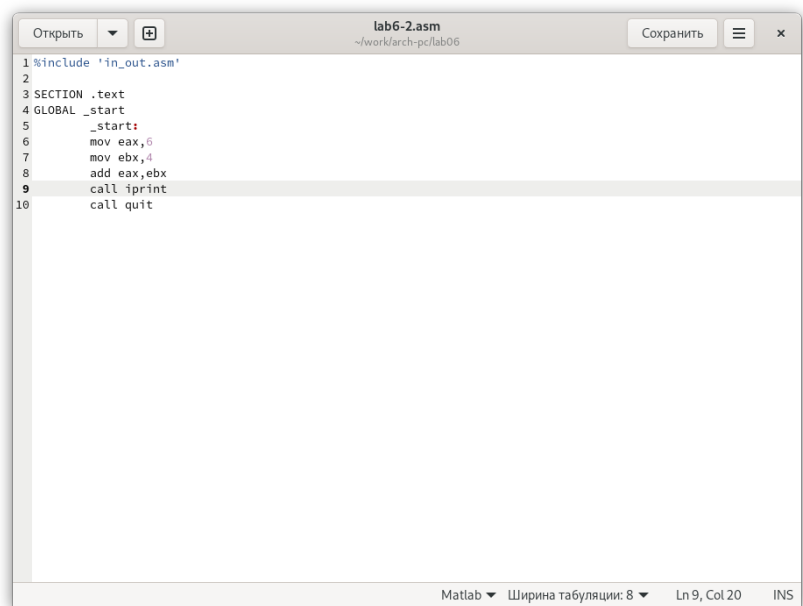


The screenshot shows a terminal window with the following commands and output:

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

Рис. 2.9: Работа программы

8. Меняем `iprintLF` на `iprint`. Вывод сработал без переноса на следующую строку.

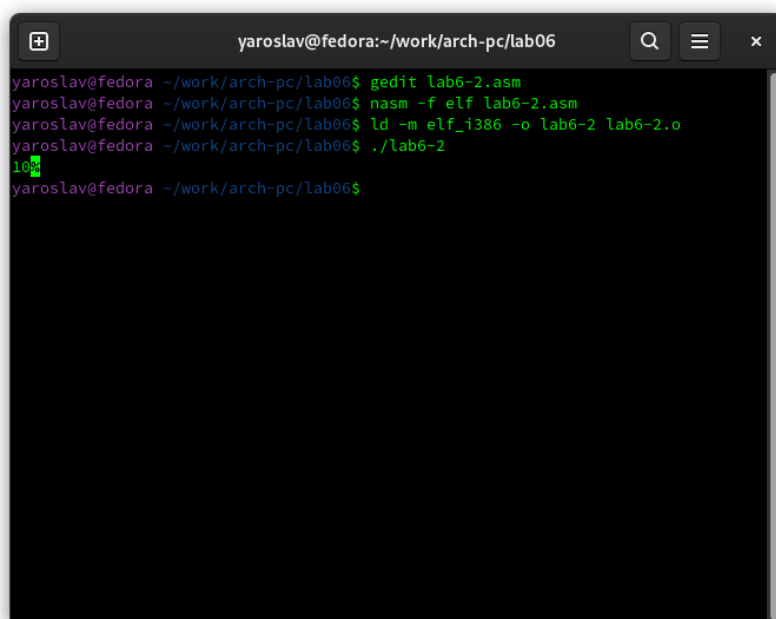


The screenshot shows a text editor window with the title bar 'lab6-2.asm' and the path '~/.work/arch-pc/lab06'. The code is as follows:

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

The status bar at the bottom indicates 'Matlab', 'Ширина табуляции: 8', 'Ln 9, Col 20', and 'INS'.

Рис. 2.10: Изменённый lab6-2.asm



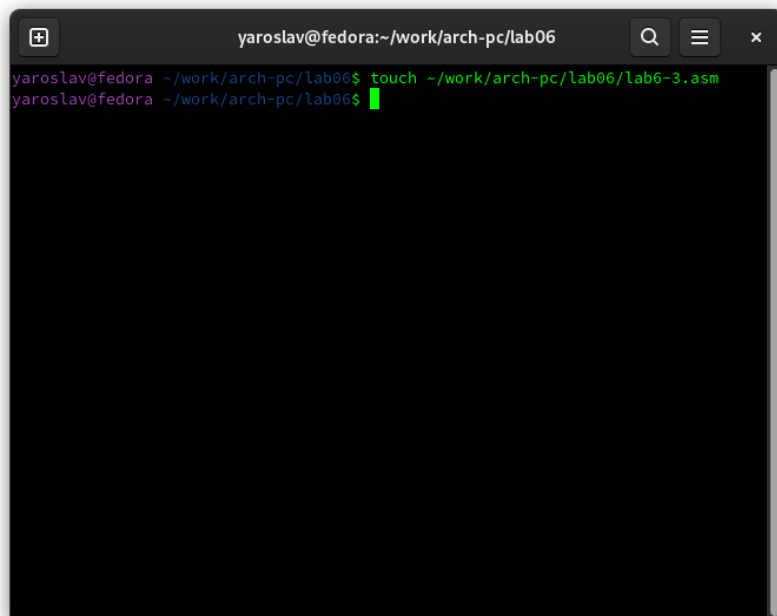
The screenshot shows a terminal window with the title bar 'yaroslav@fedora: ~/.work/arch-pc/lab06'. The commands and output are as follows:

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

Рис. 2.11: Работа программы

2.2 Выполнение арифметических операций в NASM

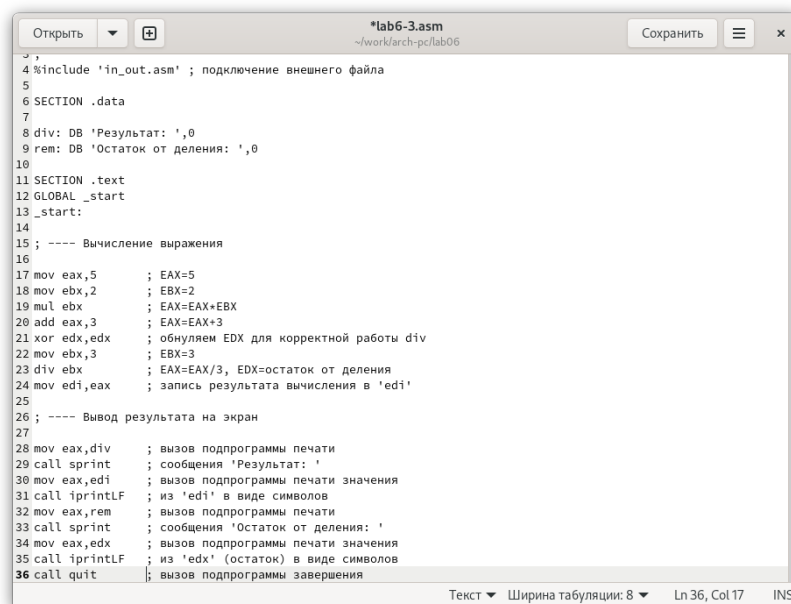
9. Создаём lab6-3.asm.

A terminal window with a dark background. The title bar shows 'yaroslav@fedora:~/work/arch-pc/lab06'. The terminal contains two lines of text: 'yaroslav@fedora ~/work/arch-pc/lab06\$ touch ~/work/arch-pc/lab06/lab6-3.asm' and 'yaroslav@fedora ~/work/arch-pc/lab06\$' followed by a green cursor. The window has standard Linux window controls (minimize, maximize, close) and a search icon.

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

Рис. 2.12: Создание файла

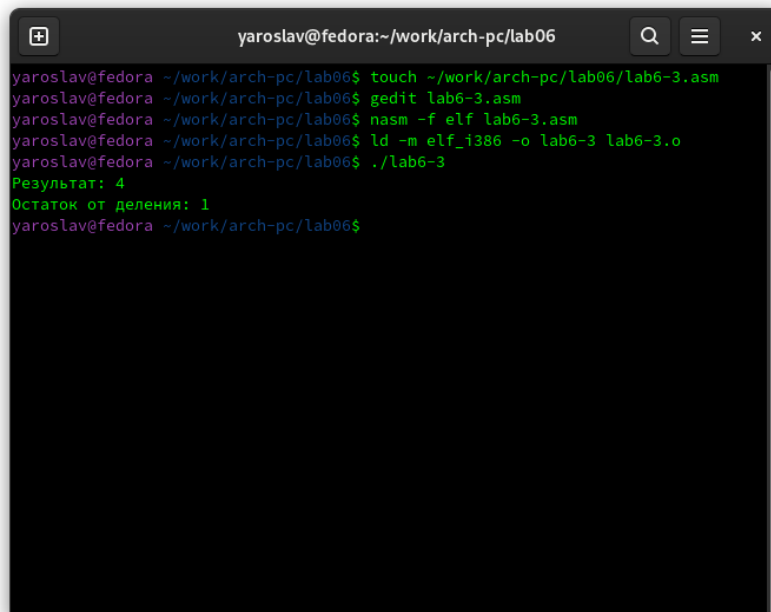
10. Копируем листинг в созданный файл.



```
1 ;
2
3 4 %include 'in_out.asm' ; подключение внешнего файла
5
6 6 SECTION .data
7
8 8 div: DB 'Результат: ',0
9 9 rem: DB 'Остаток от деления: ',0
10
11 11 SECTION .text
12 12 GLOBAL _start
13 13 _start:
14
15 15 ; ---- Вычисление выражения
16
17 17 mov eax,5 ; EAX=5
18 18 mov ebx,2 ; EBX=2
19 19 mul ebx ; EAX=EAX*EBX
20 20 add eax,3 ; EAX=EAX+3
21 21 xor edx,edx ; обнуляем EDX для корректной работы div
22 22 mov ebx,3 ; EBX=3
23 23 div ebx ; EAX=EAX/3, EDX=остаток от деления
24 24 mov edi,eax ; запись результата вычисления в 'edi'
25
26 26 ; ---- Вывод результата на экран
27
28 28 mov eax,div ; вызов подпрограммы печати
29 29 call sprint ; сообщения 'Результат: '
30 30 mov eax,edi ; вызов подпрограммы печати значения
31 31 call iprintLF ; из 'edi' в виде символов
32 32 mov eax,rem ; вызов подпрограммы печати
33 33 call sprint ; сообщения 'Остаток от деления: '
34 34 mov eax,edx ; вызов подпрограммы печати значения
35 35 call iprintLF ; из 'edx' (остаток) в виде символов
36 36 call quit ; вызов подпрограммы завершения
```

Рис. 2.13: Готовый файл lab6-3.asm

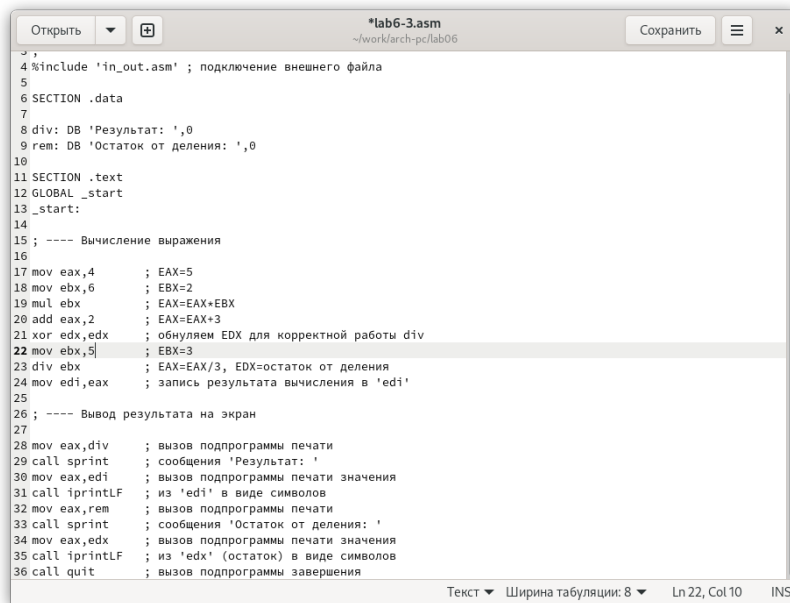
11. Запускаем, получаем результат.



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

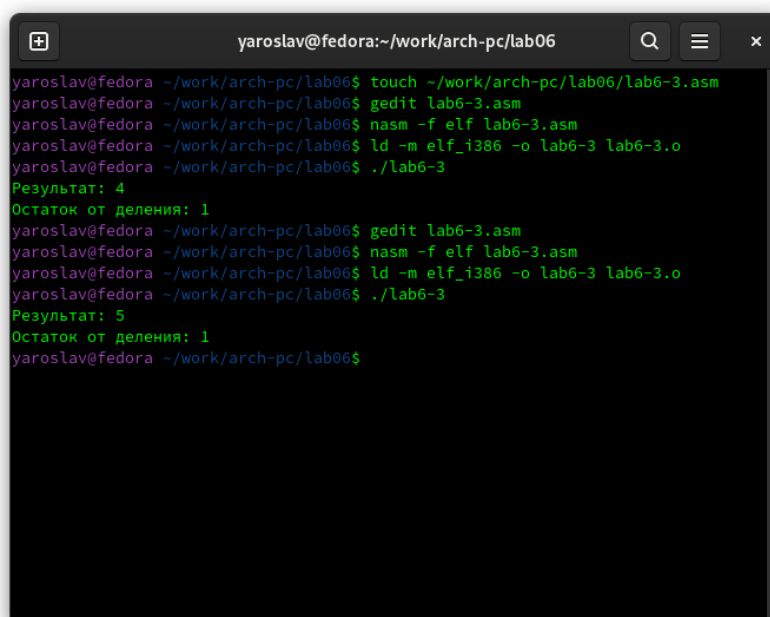
Рис. 2.14: Работа программы

12. Меняем текст программы под новые условия. Запускаем.



```
1 ;
2
3 4 %include 'in_out.asm' ; подключение внешнего файла
5
6 6 SECTION .data
7
8 8 div: DB 'Результат: ',0
9 9 rem: DB 'Остаток от деления: ',0
10
11 11 SECTION .text
12 12 GLOBAL _start
13 13 _start:
14
15 15 ; ---- Вычисление выражения
16
17 17 mov eax,4 ; EAX=5
18 18 mov ebx,6 ; EBX=2
19 19 mul ebx ; EAX=EAX*EBX
20 20 add eax,2 ; EAX=EAX+3
21 21 xor edx,edx ; обнуляем EDX для корректной работы div
22 22 mov ebx,5 ; EBX=3
23 23 div ebx ; EAX=EAX/3, EDX=остаток от деления
24 24 mov edi,eax ; запись результата вычисления в 'edi'
25
26 26 ; ---- Вывод результата на экран
27
28 28 mov eax,div ; вызов подпрограммы печати
29 29 call sprint ; сообщения 'Результат: '
30 30 mov eax,edi ; вызов подпрограммы печати значения
31 31 call iprintfLF ; из 'edi' в виде символов
32 32 mov eax,rem ; вызов подпрограммы печати
33 33 call sprint ; сообщения 'Остаток от деления: '
34 34 mov eax,edx ; вызов подпрограммы печати значения
35 35 call iprintfLF ; из 'edx' (остаток) в виде символов
36 36 call quit ; вызов подпрограммы завершения
```

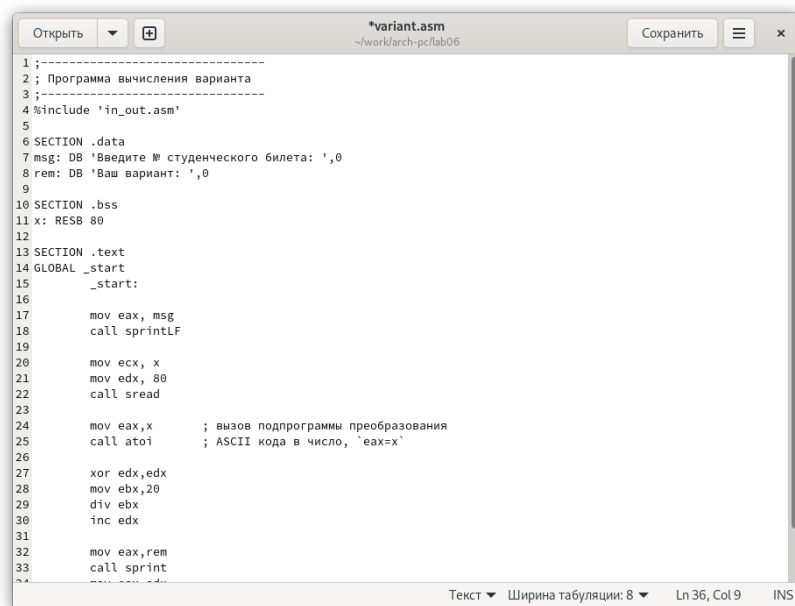
Рис. 2.15: Изменённый файл lab6-3.asm



```
yaroslav@fedora ~/work/arch-pc/lab06
yaroslav@fedora ~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
yaroslav@fedora ~/work/arch-pc/lab06$ gedit lab6-3.asm
yaroslav@fedora ~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
yaroslav@fedora ~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
yaroslav@fedora ~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
yaroslav@fedora ~/work/arch-pc/lab06$ gedit lab6-3.asm
yaroslav@fedora ~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
yaroslav@fedora ~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
yaroslav@fedora ~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
yaroslav@fedora ~/work/arch-pc/lab06$
```

Рис. 2.16: Работа программы

13. Создаём файл variant.asm и вставляем туда листинг 6.4. Запускаем и вводим номер студ.билета. Получаем номер варианта.

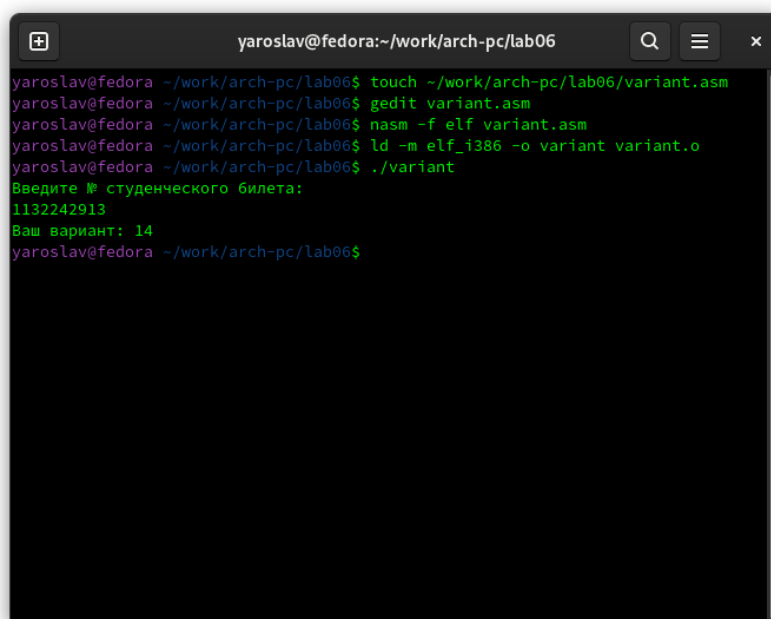


The screenshot shows a text editor window titled '*variant.asm' with the path '~/.work/arch-pc/lab06'. The code is as follows:

```
1 ;-----
2 ; Программа вычисления варианта
3 ;-----
4 %include 'in_out.asm'
5
6 SECTION .data
7 msg: DB 'Введите № студенческого билета: ',0
8 rem: DB 'Ваш вариант: ',0
9
10 SECTION .bss
11 x: RESB 80
12
13 SECTION .text
14 GLOBAL _start
15 _start:
16
17     mov eax, msg
18     call sprintf
19
20     mov ecx, x
21     mov edx, 80
22     call sread
23
24     mov eax, x      ; вызов подпрограммы преобразования
25     call atoi      ; ASCII кода в число, 'eax=x'
26
27     xor edx, edx
28     mov ebx, 20
29     div ebx
30     inc edx
31
32     mov eax, rem
33     call sprint
```

The status bar at the bottom indicates 'Текст', 'Ширина табуляции: 8', 'Ln 36, Col 9', and 'INS'.

Рис. 2.17: Файл variant.asm



The screenshot shows a terminal window with the user 'yaroslav@fedora' in the directory '~/.work/arch-pc/lab06'. The commands and output are as follows:

```
yaroslav@fedora ~/.work/arch-pc/lab06$ touch ~/.work/arch-pc/lab06/variant.asm
yaroslav@fedora ~/.work/arch-pc/lab06$ gedit variant.asm
yaroslav@fedora ~/.work/arch-pc/lab06$ nasm -f elf variant.asm
yaroslav@fedora ~/.work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
yaroslav@fedora ~/.work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132242913
Ваш вариант: 14
yaroslav@fedora ~/.work/arch-pc/lab06$
```

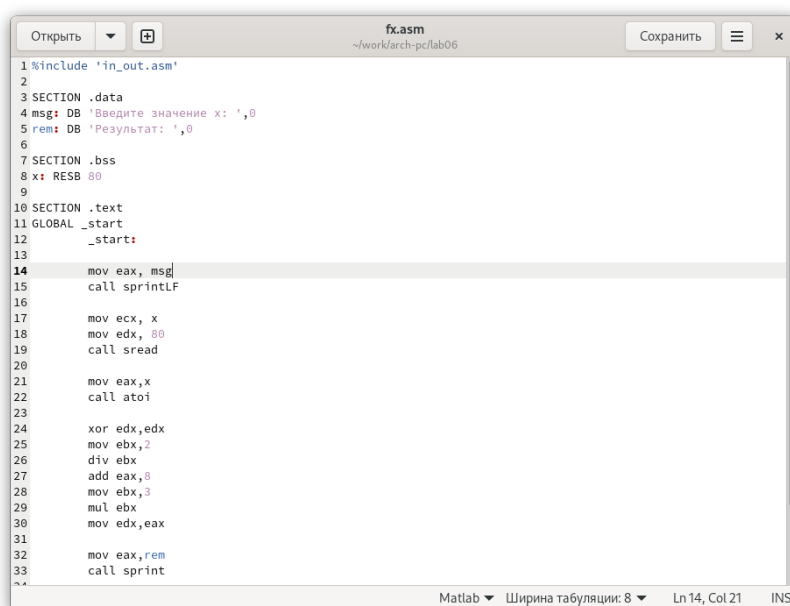
Рис. 2.18: Работа программы

2.3 Ответы на вопросы

1. Строки `mov eax,rem` и `call sprint` отвечают за вывод на экран сообщения 'Ваш вариант:'.
2. `mov ecx, x` ; запись в `eax` значения `x` `mov edx, 80` ; запись в `edx` значения 80
`call sread` ; вызов функции чтения
3. Вызов функции `atoi` - функции преобразования `ascii`-кода символа в целое число.
4. Строки `xor edx,edx` `mov ebx,20` `div ebx` `inc edx` отвечают за вычисления варианта.
5. Остаток от деления при выполнении инструкции "`div ebx`" записывается в `edx`.
6. `inc edx` увеличивает значение в `edx` на 1 и перезаписывает его в `edx`.
7. Строки `mov eax,edx` `call iprintLF` отвечают за вывод результата.

3 Выполнение самостоятельной работы

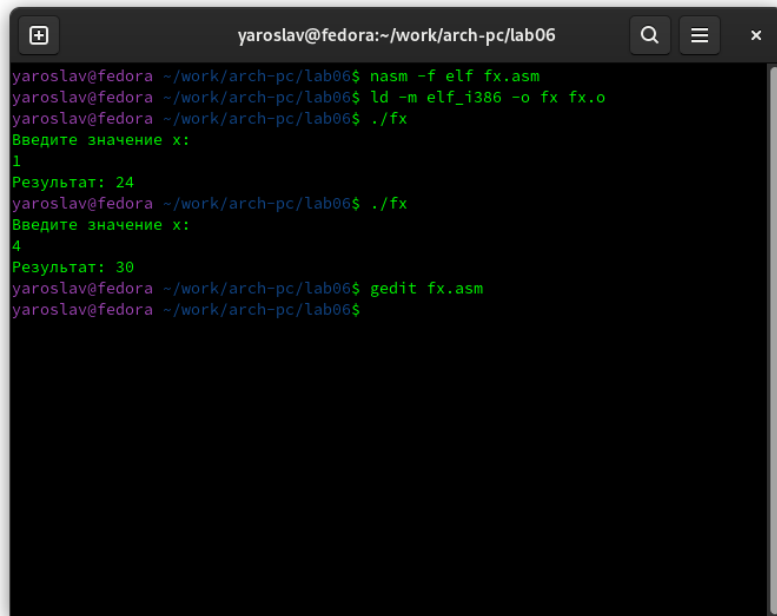
1. Создаём файл fx.asm и пишем в нём текст программы (14 вариант).



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg: DB 'Введите значение x: ',0
5 rem: DB 'Результат: ',0
6
7 SECTION .bss
8 x: RESB 80
9
10 SECTION .text
11 GLOBAL _start
12 _start:
13
14 mov eax, msg
15 call sprintf
16
17 mov ecx, x
18 mov edx, 80
19 call sread
20
21 mov eax, x
22 call atoi
23
24 xor edx, edx
25 mov ebx, 2
26 div ebx
27 add eax, 8
28 mov ebx, 3
29 mul ebx
30 mov edx, eax
31
32 mov eax, rem
33 call sprint
```

Рис. 3.1: Готовый fx.asm

2. Проверяем работу на заданных числах.

A terminal window titled 'yaroslav@fedora:~/work/arch-pc/lab06' with search and menu icons. It shows the execution of assembly and linking commands, followed by two runs of a program that prompts for input 'x' and outputs results. The first run takes input '1' and outputs '24'. The second run takes input '4' and outputs '30'. The window ends with the command 'gedit fx.asm' and the prompt 'yaroslav@fedora:~/work/arch-pc/lab06\$'.

```
yaroslav@fedora ~/work/arch-pc/lab06$ nasm -f elf fx.asm
yaroslav@fedora ~/work/arch-pc/lab06$ ld -m elf_i386 -o fx fx.o
yaroslav@fedora ~/work/arch-pc/lab06$ ./fx
Введите значение x:
1
Результат: 24
yaroslav@fedora ~/work/arch-pc/lab06$ ./fx
Введите значение x:
4
Результат: 30
yaroslav@fedora ~/work/arch-pc/lab06$ gedit fx.asm
yaroslav@fedora ~/work/arch-pc/lab06$
```

Рис. 3.2: Работа программы

4 Выводы

Были освоены арифметические операции в NASM