

Отчет по лабораторной работе №6

Дисциплина: архитектура компьютера

Хамзина Виктория Валентиновна

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	10
2.3	Ответы на вопросы	13
2.4	Задание для самостоятельной работы	14
3	Выводы	17

Список иллюстраций

2.1	Создание каталога и файла	5
2.2	Копирование файла	5
2.3	Редактирование файла	6
2.4	Запуск исполняемого файла	6
2.5	Редактирование файлов	7
2.6	Запуск исполняемого файла	7
2.7	Создание файла	7
2.8	Редактирование файла	8
2.9	Запуск исполняемого файла	8
2.10	Редактирование файла	9
2.11	Запуск исполняемого файла	9
2.12	Редактирование файла	10
2.13	Запуск исполняемого файла	10
2.14	Создание файла	10
2.15	Редактирование файла	11
2.16	Запуск исполняемого файла	11
2.17	Редактирование файла	12
2.18	Запуск исполняемого файла	12
2.19	Создание файла	12
2.20	Редактирование файла	13
2.21	Запуск исполняемого файла	13
2.22	Создание файла	14
2.23	Редактирование файла	15
2.24	Запуск исполняемого файла	16

1 Цель работы

Освоить арифметические инструкции языка ассемблера NASM.

2 Выполнение лабораторной работы

2.1 Символьные и численные данные в NASM

Создала каталог для программ лабораторной работы №6 и перешла в него. Создала файл lab6-1.asm с помощью команды touch (рис. 2.1).

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

Рис. 2.1: Создание каталога и файла

Скопировала подключаемый файл in_out.asm в каталог ~/work/arch-pc/lab06 из директории ~/Загрузки для корректной работы программы с помощью файлового менеджера Midnight Commander (рис. 2.2).

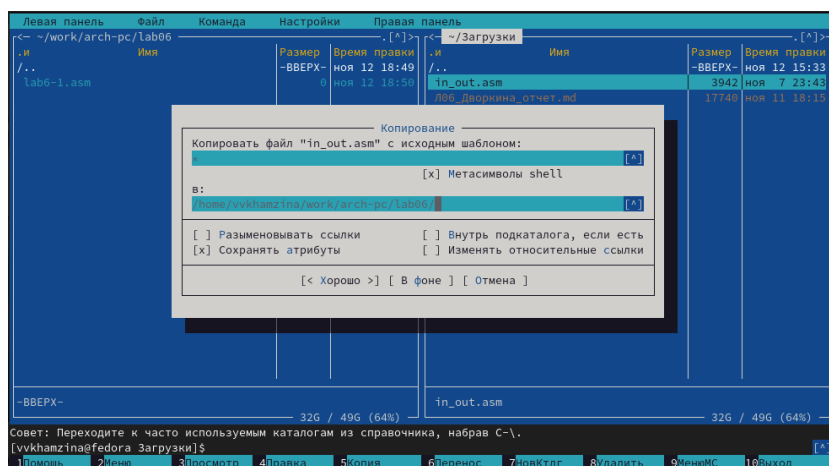
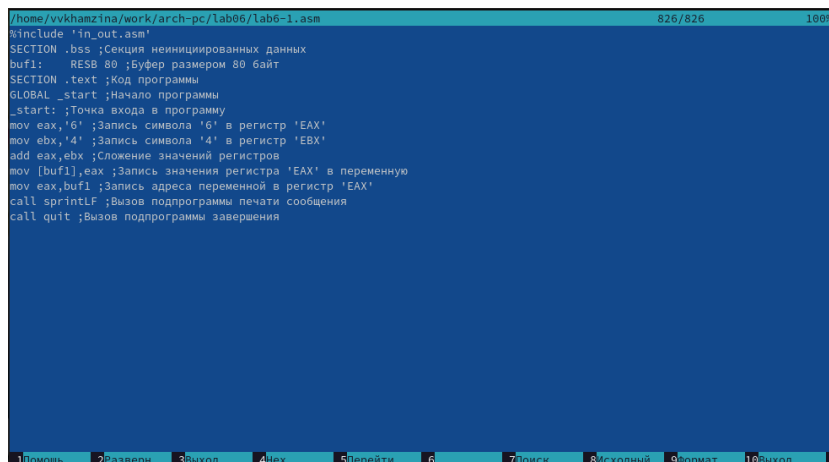


Рис. 2.2: Копирование файла

Открыла файл lab6-1.asm и ввела в него текст программы вывода значения регистра eax (рис. 2.3).



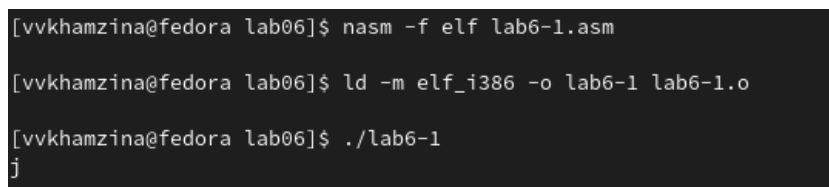
```

/home/vvkhamzina/work/arch-pc/lab06/lab6-1.asm  826/826  100%
%include 'in_out.asm'
SECTION .bss ;Секция неиницированных данных
buf1:  RESB 80 ;Буфер размером 80 байт
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,'6' ;Запись символа '6' в регистр 'EAX'
mov ebx,'4' ;Запись символа '4' в регистр 'EBX'
add eax,ebx ;Сложение значений регистров
mov [buf1],eax ;Запись значения регистра 'EAX' в переменную
mov eax,buf1 ;Запись адреса переменной в регистр 'EAX'
call sprintf ;Вызов подпрограммы печати сообщения
call quit ;Вызов подпрограммы завершения

```

Рис. 2.3: Редактирование файла

Создала исполняемый файл lab6-1 и запустила его. Программа вывела j, так как этот символ соответствует сумме двоичных кодов символов 4 и 6 в системе ASCII (рис. 2.4).



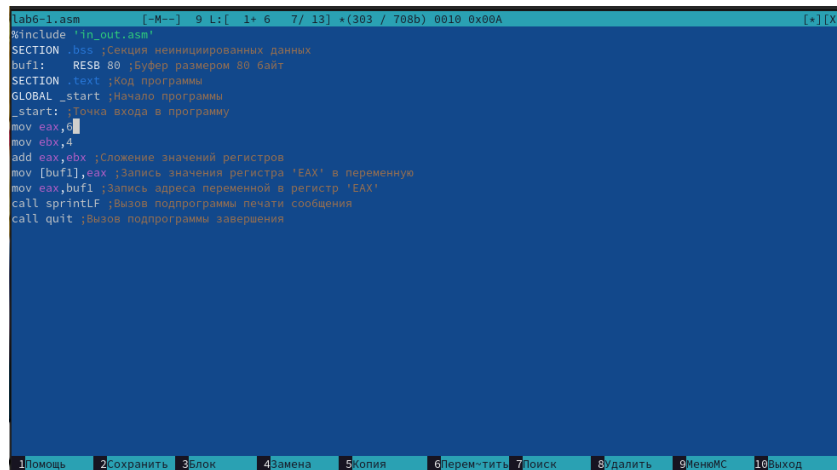
```

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

```

Рис. 2.4: Запуск исполняемого файла

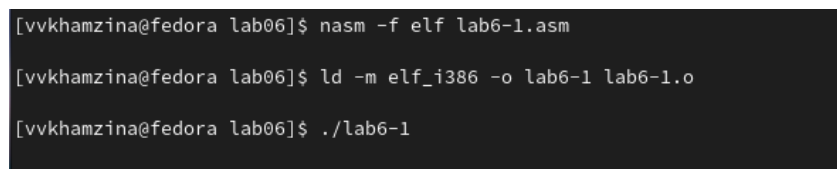
Изменила текст программы, записав в регистры eax и ebx вместо символов '6' и '4' цифры 6 и 4 (рис. 2.5).



```
lab6-1.asm [-M--] 9 L: [ 1+ 6 7/ 13] *(303 / 708b) 0010 0x00A [*][X]
%include 'in_out.asm'
SECTION .bss ;Секция неиницированных данных
buf1: RESB 80 ;Буфер размером 80 байт
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,6
mov ebx,4
add eax,ebx ;Сложение значений регистров
mov [buf1],eax ;Запись значения регистра 'EAX' в переменную
mov eax,buf1 ;Запись адреса переменной в регистр 'EAX'
call sprintf ;Вызов подпрограммы печати сообщения
call quit ;Вызов подпрограммы завершения
```

Рис. 2.5: Редактирование файла

Создала исполняемый файл после изменений текста программы и запустила его. Программа вывела символ с кодом 10 - символ перевода строки, который не отображается на экране (рис. 2.6).



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

Рис. 2.6: Запуск исполняемого файла

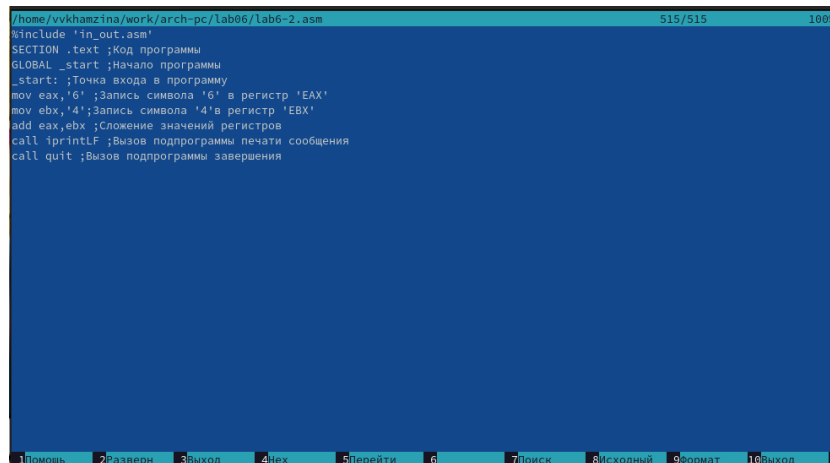
Создала файл lab6-2.asm в директории ~/work/arch-pc/lab06 с помощью утилиты touch (рис. 2.7).



```
[vvkhamzina@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm
```

Рис. 2.7: Создание файла

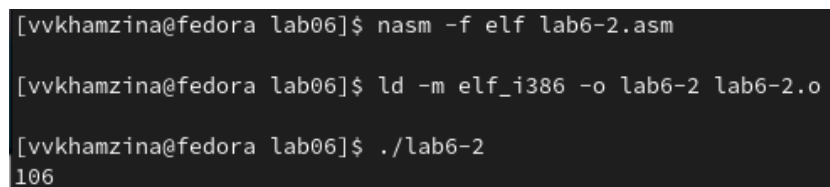
Ввела в созданный файл другой текст программы для вывода значения регистра eax (рис. 2.8).



```
/home/vvkhazina/work/arch-pc/lab06/lab6-2.asm 515/515 100%
#include 'in_out.asm'
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,'6' ;Запись символа '6' в регистр 'EAX'
mov ebx,'4' ;Запись символа '4' в регистр 'EBX'
add eax,ebx ;Сложение значений регистров
call iprintLF ;Вызов подпрограммы печати сообщения
call quit ;Вызов подпрограммы завершения
```

Рис. 2.8: Редактирование файла

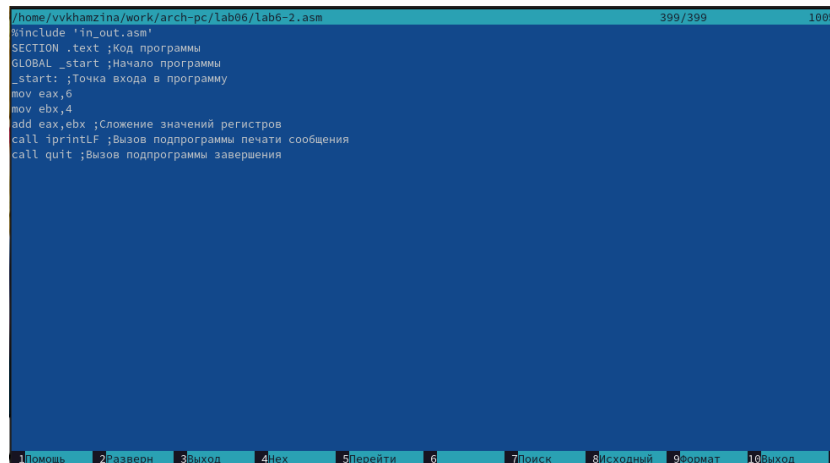
Создала исполняемый файл и запустила его. Программа вывела число 106. Как и в прошлый раз это является суммой двоичных кодов 6 и 4, но функция `iprintLF` вывела уже число, а не символ, кодом которого является это число (рис. 2.9).



```
[vvkhamzina@fedora lab06]$ nasm -f elf lab6-2.asm
[vvkhamzina@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[vvkhamzina@fedora lab06]$ ./lab6-2
106
```

Рис. 2.9: Запуск исполняемого файла

Изменила текст программы файла `lab6-1`, записав в регистры `eax` и `ebx` вместо символов '6' и '4' цифры 6 и 4 (рис. 2.10).

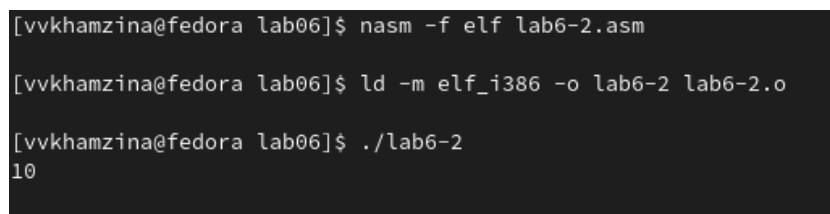


```
/home/vvkhamzina/work/arch-pc/lab06/lab6-2.asm 399/399 100%
%include 'in_out.asm'
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,6
mov ebx,4
add eax,ebx ;Сложение значений регистров
call iprintf ;Вызов подпрограммы печати сообщения
call quit ;Вызов подпрограммы завершения
```

1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход

Рис. 2.10: Редактирование файла

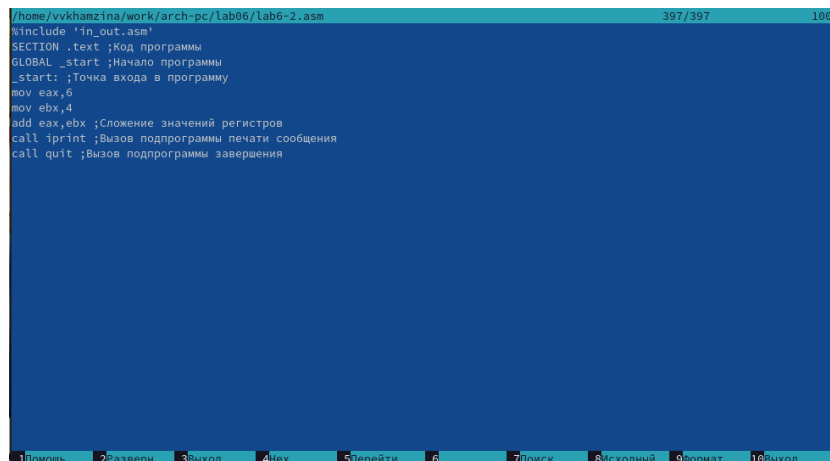
Создала исполняемый файл и запустила его. Программа вывела число 10, так как сложила сами числа, а не их двоичные коды (рис. 2.11).



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

Рис. 2.11: Запуск исполняемого файла

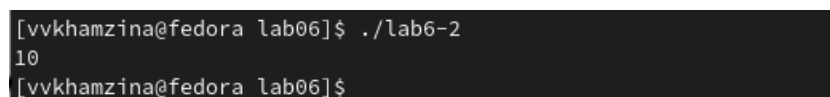
Заменяла в тексте программы функцию iprintLF на iprint (рис. 2.12).



```
/home/vvkhazina/work/arch-pc/lab06/lab6-2.asm 397/397 100%
#include "in_out.asm"
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,6
mov ebx,4
add eax,ebx ;Сложение значений регистров
call iprint ;Вызов подпрограммы печати сообщения
call quit ;Вызов подпрограммы завершения
```

Рис. 2.12: Редактирование файла

Создала исполняемый файл после изменений текста программы и запустила его. Теперь программа не переводит строку после вывода 10 (рис. 2.13).



```
[vvkhazina@fedora lab06]$ ./lab6-2
10
[vvkhazina@fedora lab06]$
```

Рис. 2.13: Запуск исполняемого файла

2.2 Выполнение арифметических операций в NASM

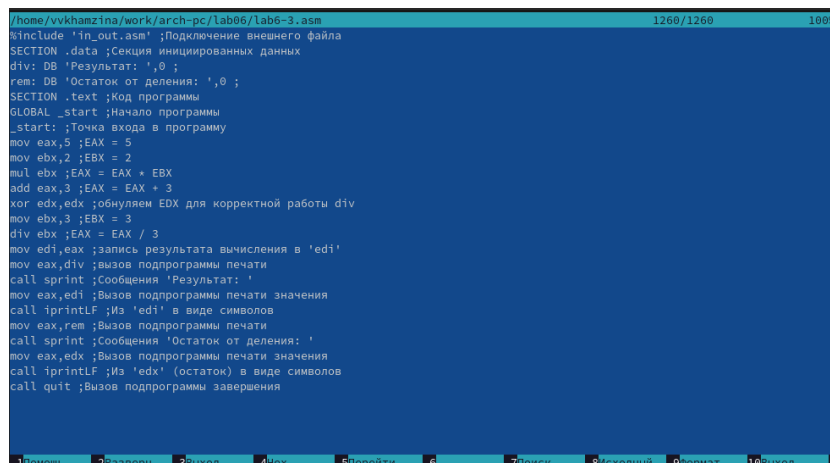
Создала файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 2.14).



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

Рис. 2.14: Создание файла

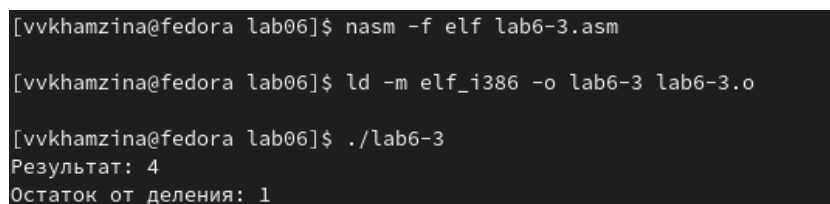
Ввела в созданный файл программу вычисления выражения $f(x)=(5*2+3)/3$ (рис. 2.15).



```
/home/vvkhamzina/work/arch-pc/lab06/lab6-3.asm 1260/1260 100%
#include 'in_out.asm' ;Подключение внешнего файла
SECTION .data ;Секция инициализированных данных
div: DB 'Результат: ',0 ;
rem: DB 'Остаток от деления: ',0 ;
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
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 / 3
mov edi,eax ;запись результата вычисления в 'edi'
mov eax,div ;вызов подпрограммы печати
call sprint ;Сообщения 'Результат: '
mov eax,edi ;Вызов подпрограммы печати значения
call iprintf ;Из 'edi' в виде символов
mov eax,rem ;Вызов подпрограммы печати
call sprint ;Сообщения 'Остаток от деления: '
mov eax,edx ;Вызов подпрограммы печати значения
call iprintf ;Из 'edx' (остаток) в виде символов
call quit ;Вызов подпрограммы завершения
```

Рис. 2.15: Редактирование файла

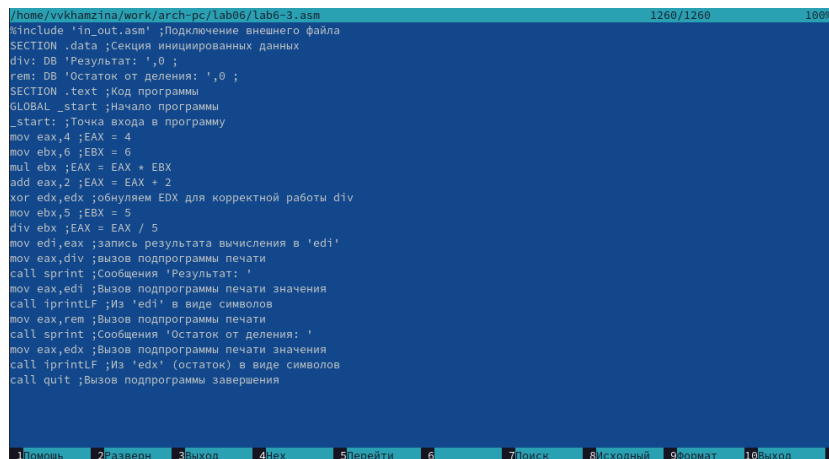
Создала исполняемый файл и запустила его (рис. 2.16).



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

Рис. 2.16: Запуск исполняемого файла

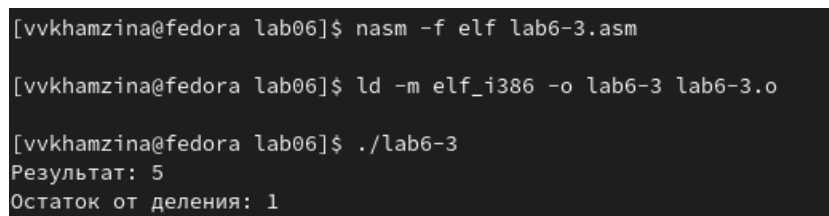
Изменила текст программы файла lab6-3.asm так, чтобы программы вычисляла значение выражения $f(x)=(4*6+2)/5$ (рис. 2.17).



```
//home/vvkhamzina/work/arch-pc/lab06/lab6-3.asm
#include "in_out.asm" ;подключение внешнего файла
SECTION .data ;соединяю инициализированных данных
div: DB 'Результат: ',0 ;
rem: DB 'Остаток от деления: ',0 ;
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,4 ;EAX = 4
mov ebx,6 ;EBX = 6
mul ebx ;EAX = EAX * EBX
add eax,2 ;EAX = EAX + 2
xor edx,edx ;обнуляем EDX для корректной работы div
mov ebx,5 ;EBX = 5
div ebx ;EAX = EAX / 5
mov edi,eax ;запись результата вычисления в 'edi'
mov eax,div ;вызов подпрограммы печати
call sprint ;Сообщения 'Результат: '
mov eax,edi ;Вызов подпрограммы печати значения
call 'printf' ;из 'edi' в виде символов
mov eax,rem ;Вызов подпрограммы печати
call sprint ;Сообщения 'Остаток от деления: '
mov eax,edx ;Вызов подпрограммы печати значения
call 'printf' ;из 'edx' (остаток) в виде символов
call quit ;Вызов подпрограммы завершения
```

Рис. 2.17: Редактирование файла

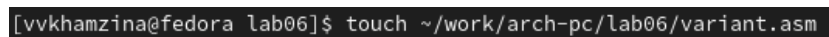
Создала исполняемый файл после изменений текста программы и запустила его (рис. 2.18).



```
[vvkhamzina@fedora lab06]$ nasm -f elf lab6-3.asm
[vvkhamzina@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[vvkhamzina@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 2.18: Запуск исполняемого файла

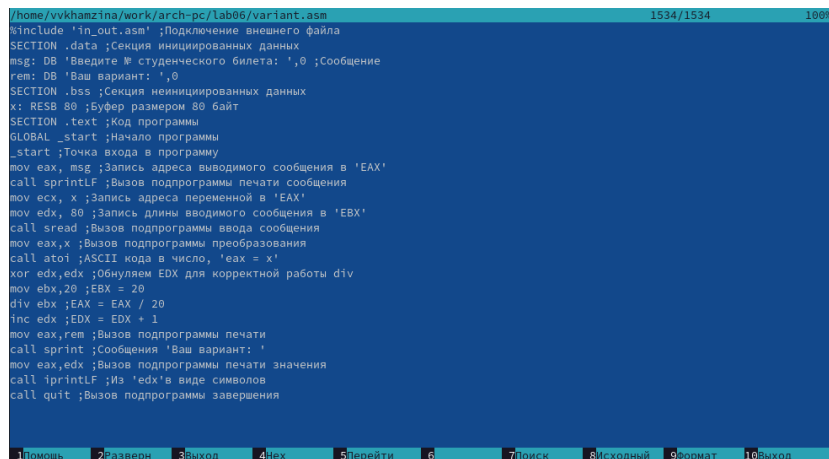
Создала файл variant.asm с помощью утилиты touch (рис. 2.19).



```
[vvkhamzina@fedora lab06]$ touch ~/work/arch-pc/lab06/variant.asm
```

Рис. 2.19: Создание файла

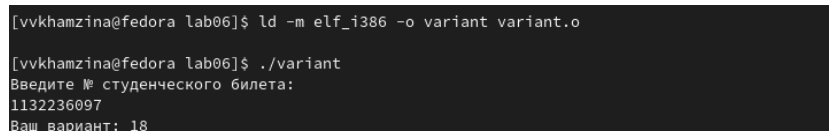
Ввела в созданный файл программу для вычисления варианта задания по номеру студенческого билета (рис. 2.20).



```
/home/vvkhazina/work/arch-pc/lab06/variant.asm
#include "in_out.asm" ;Подключение внешнего файла
SECTION .data ;Секция инициализированных данных
msg: DB 'Введите № студенческого билета: ',0 ;Сообщение
rem: DB 'Ваш вариант: ',0
SECTION .bss ;Секция неинициализированных данных
x: RESB 80 ;Буфер размером 80 байт
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start ;Точка входа в программу
mov eax, msg ;Запись адреса выводимого сообщения в 'EAX'
call sprintf ;Вызов подпрограммы печати сообщения
mov ecx, x ;Запись адреса переменной в 'EAX'
mov edx, 80 ;Запись длины вводимого сообщения в 'EBX'
call sread ;Вызов подпрограммы ввода сообщения
mov eax,x ;Вызов подпрограммы преобразования
call atoi ;ASCII кода в число, 'eax = x'
xor edx,edx ;Обнуляем EDX для корректной работы div
mov ebx,20 ;EBX = 20
div ebx ;EAX = EAX / 20
inc edx ;EDX = EDX + 1
mov eax,rem ;Вызов подпрограммы печати
call sprintf ;Сообщения 'Ваш вариант: '
mov eax,edx ;Вызов подпрограммы печати значения
call printf ;Из 'edx' в виде символов
call quit ;Вызов подпрограммы завершения
```

Рис. 2.20: Редактирование файла

Создала и запустила исполняемый файл variant (рис. 2.21).



```
[vvkhazina@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[vvkhazina@fedora lab06]$ ./variant
Введите № студенческого билета:
1132236097
Ваш вариант: 18
```

Рис. 2.21: Запуск исполняемого файла

2.3 Ответы на вопросы

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

```
mov eax, rem
call sprintf
```

2. Инструкция 'mov ecx,x' используется для записи адреса вводимой переменной x в регистр ecx, 'mov edx, 80' - для записи длины вводимого сообщения в регистр edx. 'call sread' вызывает подпрограмму ввода сообщения с клавиатуры.
3. 'Call atoi' используется для преобразования ASCII кода символа в число и записи результата в регистр eax.

4. За вычисление варианта отвечают следующие строки:

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

5. При выполнении инструкции 'div ebx' остаток от деления записывается в регистр edx.

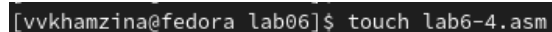
6. Инструкция 'inc edx' используется для увеличения значения, записанного в регистр edx, на 1.

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

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

2.4 Задание для самостоятельной работы

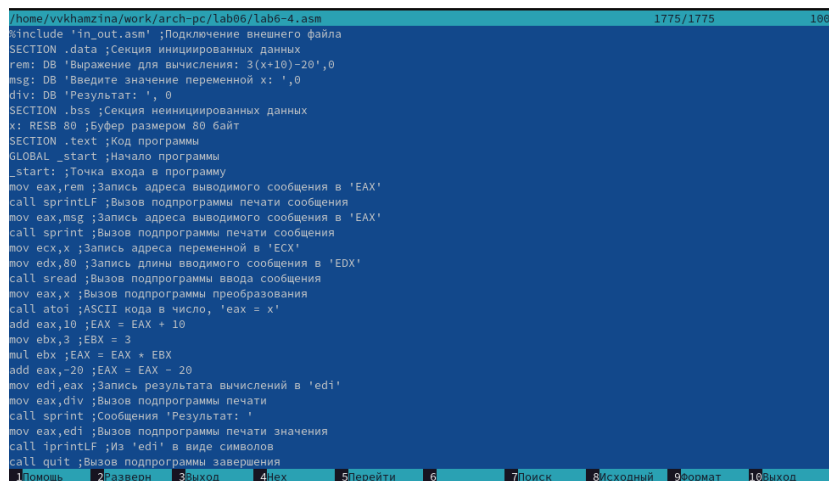
Создала файл lab6-4.asm (рис. 2.22).



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

Рис. 2.22: Создание файла

Написала текст программы для вычисления выражения под вариантом 18, так как он попался мне по номеру моего студенческого билета (рис. 2.23).



```
/home/vvkhazina/work/arch-pc/lab06/lab6-4.asm 1775/1775 100%
#include 'in_out.asm' ;Подключение внешнего файла
SECTION .data ;Секция инициированных данных
rem: DB 'Выражение для вычисления: 3(x+10)-20',0
msg: DB 'Введите значение переменной x: ',0
div: DB 'Результат: ',0
SECTION .bss ;Секция неиницированных данных
x: RESB 80 ;Буфер размером 80 байт
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,rem ;Запись адреса выводимого сообщения в 'EAX'
call sprintf ;Вызов подпрограммы печати сообщения
mov eax,msg ;Запись адреса выводимого сообщения в 'EAX'
call sprintf ;Вызов подпрограммы печати сообщения
mov ecx,x ;Запись адреса переменной в 'ECX'
mov edx,80 ;Запись длины вводимого сообщения в 'EDX'
call sread ;Вызов подпрограммы ввода сообщения
mov eax,x ;Вызов подпрограммы преобразования
call atoi ;ASCII кода в число, 'eax = x'
add eax,10 ;EAX = EAX + 10
mov ebx,3 ;EBX = 3
mul ebx ;EAX = EAX * EBX
add eax,-20 ;EAX = EAX - 20
mov edi,eax ;Запись результата вычислений в 'edi'
call sprint ;Вызов подпрограммы печати
call sprint ;Сообщения 'Результат: '
mov eax,edi ;Вызов подпрограммы печати значения
call sprintf ;%i 'edi' в виде символов
call quit ;Вызов подпрограммы завершения
1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

Рис. 2.23: Редактирование файла

Код программы для вычисления выражения 18. $3(x+10)-20$

```
%include 'in_out.asm' ;Подключение внешнего файла
SECTION .data ;Секция инициированных данных
rem: DB 'Выражение для вычисления: 3(x+10)-20',0
msg: DB 'Введите значение переменной x: ',0
SECTION .bss ;Секция неиницированных данных
buf1: RESB 80 ;Буфер размером 80 байт
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;Точка входа в программу
mov eax,rem ;Запись адреса выводимого сообщения в 'EAX'
call sprintf ;Вызов подпрограммы печати сообщения
mov eax,msg ;Запись адреса выводимого сообщения в 'EAX'
call sprintf ;Вызов подпрограммы печати сообщения
mov ecx,x ;Запись адреса переменной в 'ECX'
mov edx,80 ;Запись длины вводимого сообщения в 'EDX'
call sread ;Вызов подпрограммы ввода сообщения
mov eax,x ;Вызов подпрограммы преобразования
```

```

call atoi ;ASCII кода в число, 'eax = x'
add eax,10 ;EAX = EAX + 10
mov ebx,3 ;EBX = 3
mul ebx ;EAX = EAX * EBX
add eax,-20 ;EAX = EAX - 20
mov edi,eax ;Запись результата вычислений в 'edi'
mov eax,div ;Вызов подпрограммы печати
call sprint ;Сообщение 'Результат: '
mov eax,edi ;Вызов подпрограммы печати значения
call iprintLF ;Из 'edi' в виде символов
call quit ;Вызов подпрограммы завершения

```

Создала исполняемый файл и проверила его работу, подставив два значения x (рис. 2.24).

```

[vvkhamzina@fedora lab06]$ nasm -f elf lab6-4.asm
[vvkhamzina@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[vvkhamzina@fedora lab06]$ ./lab6-4
Выражение для вычисления: 3(x+10)-20
Введите значение переменной x: 1
Результат: 13
[vvkhamzina@fedora lab06]$ ./lab6-4
Выражение для вычисления: 3(x+10)-20
Введите значение переменной x: 5
Результат: 25
[vvkhamzina@fedora lab06]$

```

Рис. 2.24: Запуск исполняемого файла

3 Выводы

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