Отчет по лабораторной работе №6

Дисциплина: архитектура компьютера

Хамзина Виктория Валентиновна

Содержание

# 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. [1](#fig:001)).

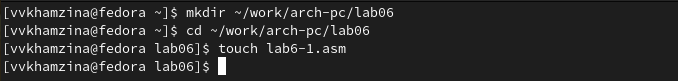


Figure 1: Создание каталога и файла

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](#fig:002)).

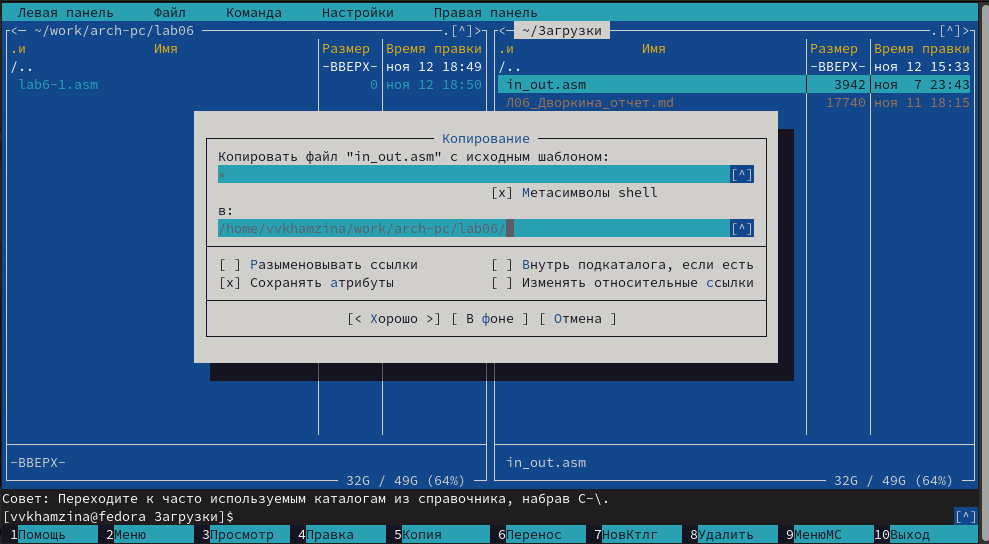


Figure 2: Копирование файла

Открыла файл lab6-1.asm и ввела в него текст программы вывода значения регистра eax (рис. [3](#fig:003)).

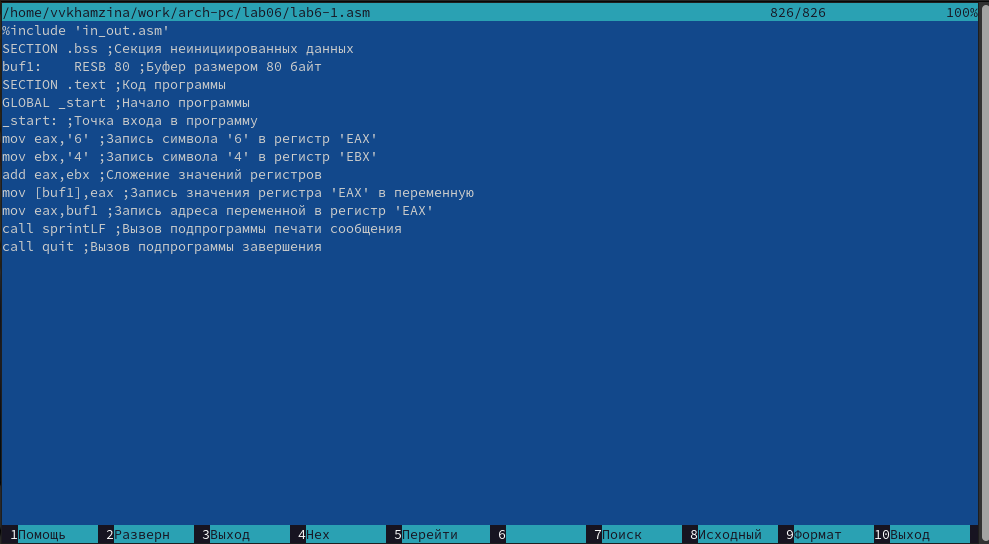


Figure 3: Редактирование файла

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. [4](#fig:004)).

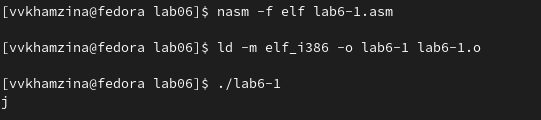


Figure 4: Запуск исполняемого файла

Изменила текст программы, записав в регистры eax и ebx вместо символов ‘6’ и ‘4’ цифры 6 и 4 (рис. [5](#fig:005)).

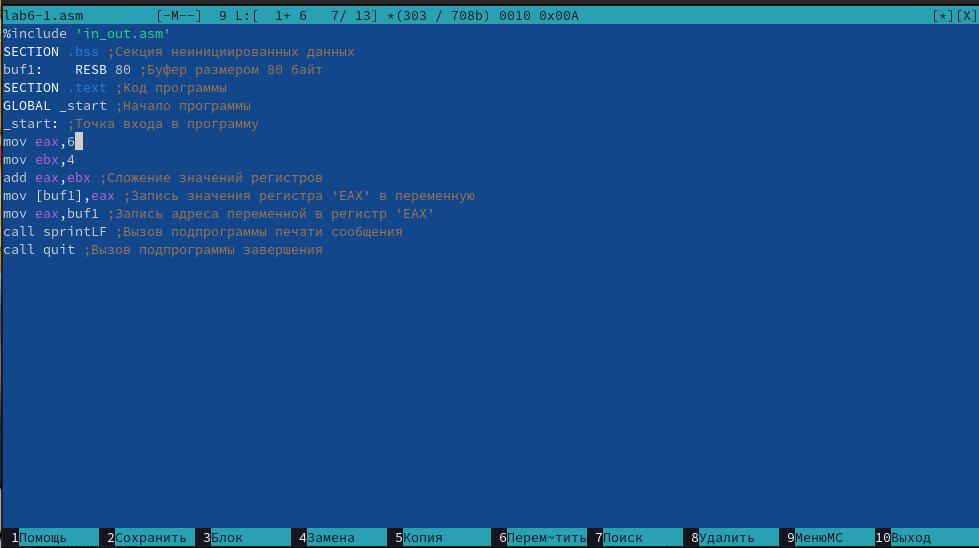


Figure 5: Редактирование файлв

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. [6](#fig:006)).

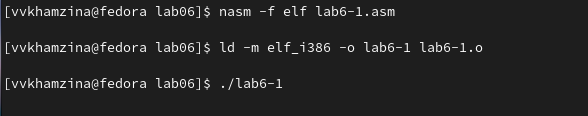


Figure 6: Запуск исполняемого файла

Создала файл lab6-2.asm в директории ~/work/arch-pc/lab06 с помощью утилиты touch (рис. [7](#fig:007)).

Figure 7: Создание файла

Figure 7: Создание файла

Ввела в созданный файл другой текст программы для вывода значения регистра eax (рис. [8](#fig:008)).

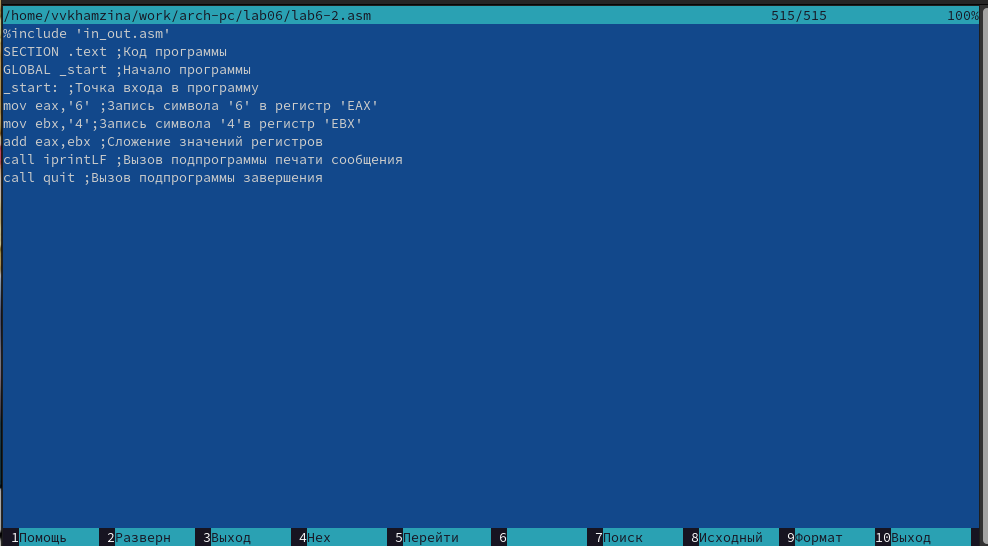


Figure 8: Редактирование файла

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. [9](#fig:009)).

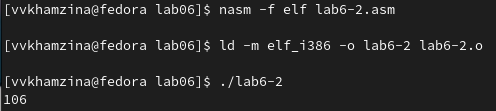


Figure 9: Запуск исполняемого файла

Изменила текст программы файла lab6-1, записав в регистры eax и ebx вместо символов ‘6’ и ‘4’ цифры 6 и 4 (рис. [10](#fig:010)).

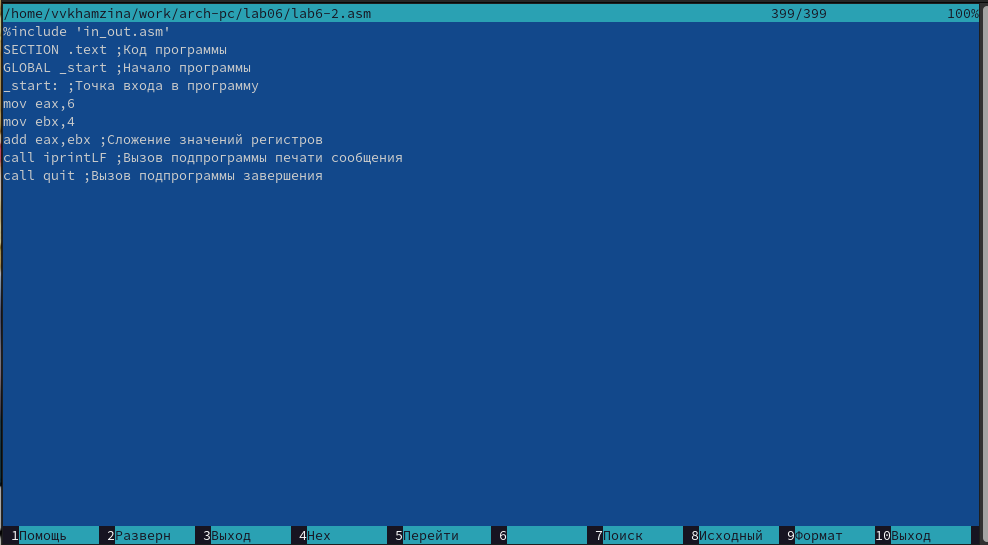


Figure 10: Редактирование файла

Создала исполняемый файл и запустила его. Программа вывела число 10, так как сложила сами числа, а не их двоичные коды (рис. [11](#fig:011)).

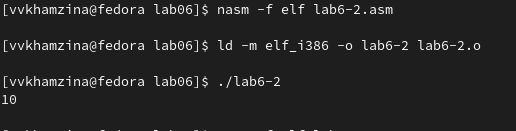


Figure 11: Запуск исполняемого файла

Заменила в тексте программы функцию iprintLF на iprint (рис. [12](#fig:012)).

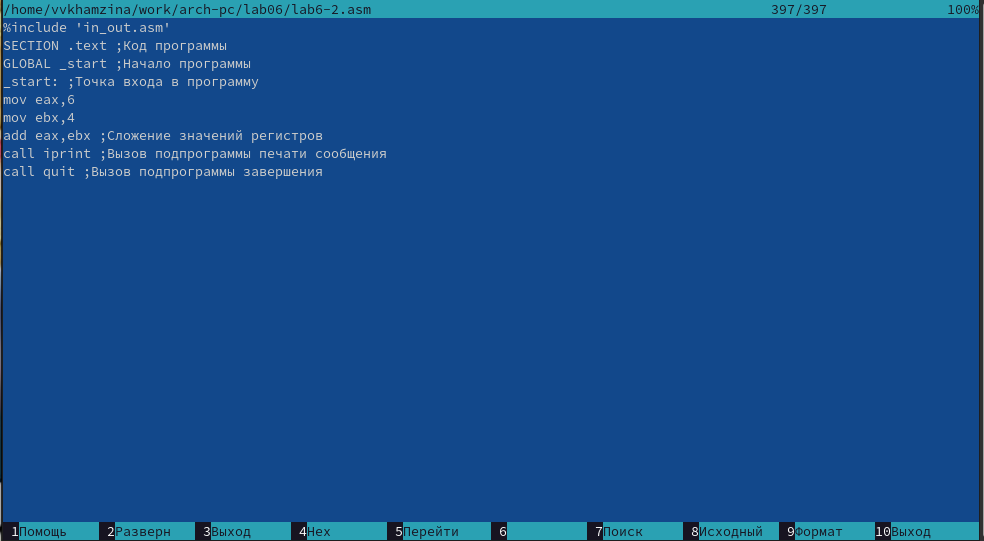


Figure 12: Редактирование файла

Создала исполняемый файл после изменений текста программы и запустила его. Теперь программа не переводит строку после вывода 10 (рис. [13](#fig:013)).

Figure 13: Запуск исполняемого файла

Figure 13: Запуск исполняемого файла

## 2.2 Выполнение арифметических операций в NASM

Создала файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. [14](#fig:014)).

Figure 14: Создание файла

Figure 14: Создание файла

Ввела в созданный файл программу вычисления выражения f(x)=(5\*2+3)/3 (рис. [15](#fig:015)).

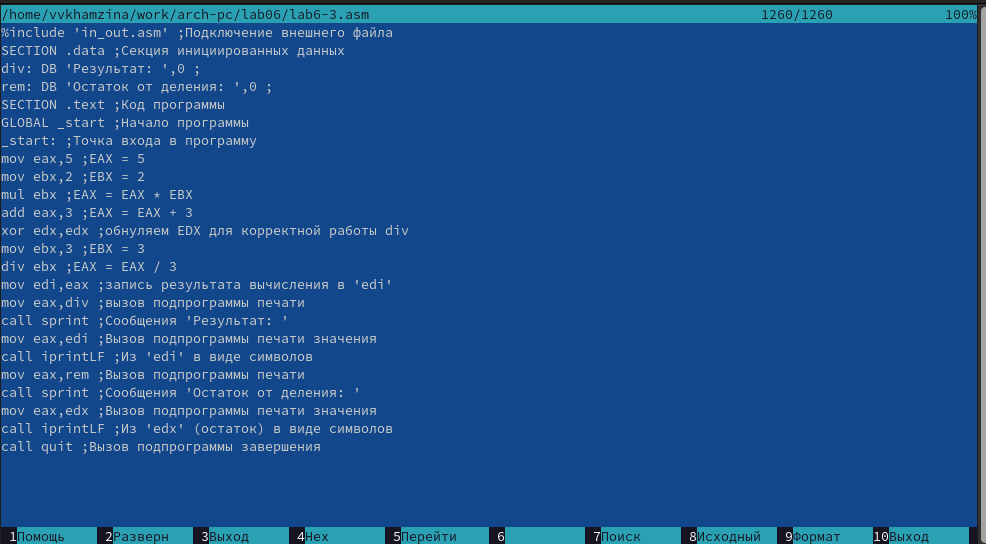


Figure 15: Редактирование файла

Создала исполняемый файл и запустила его (рис. [16](#fig:016)).

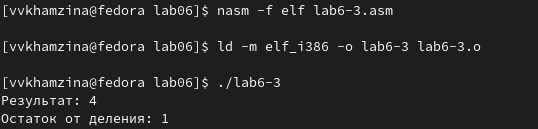


Figure 16: Запуск исполняемого файла

Изменила текст программы файла lab6-3.asm так, чтобы программы вычисляла значение выражения f(x)=(4\*6+2)/5 (рис. [17](#fig:017)).

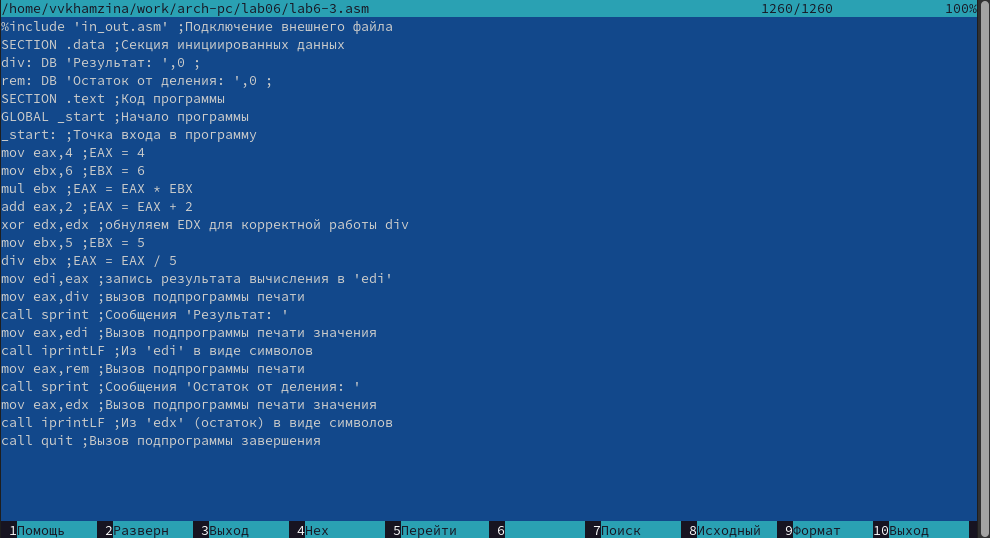


Figure 17: Редактирование файла

Создала исполняемый файл после изменений текста программы и запустила его (рис. [18](#fig:018)).

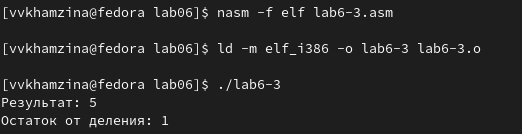


Figure 18: Запуск исполняемого файла

Создала файл variant.asm с помощью утилиты touch (рис. [19](#fig:019)).

Figure 19: Создание файла

Figure 19: Создание файла

Ввела в созданный файл программу для вычисления варианта задания по номеру студенческого билета (рис. [20](#fig:020)).

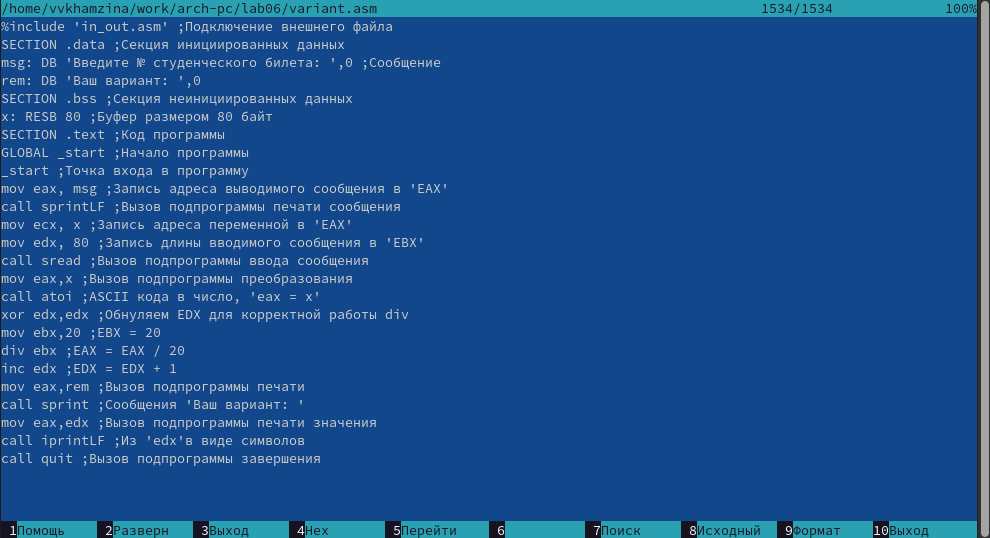


Figure 20: Редактирование файла

Создала и запустила исполняемый файл variant (рис. [21](#fig:021)).

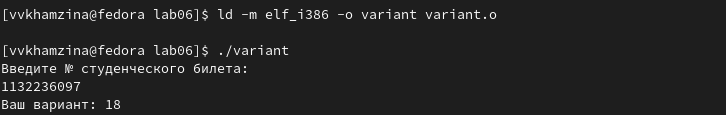


Figure 21: Запуск исполняемого файла

## 2.3 Ответы на вопросы

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

mov eax,rem  
call sprint

1. 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.
2. ‘Call atoi’ используется для преобразования ASCII кода символа в число и записи результата в регистр eax.
3. За вычисление варианта отвечают следующие строки:

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

1. При выполнении инструкции ‘div ebx’ остаток от деления записывается в регистр edx.
2. Инструкция ‘inc edx’ используется для увеличения значения, записанного в регистр edx, на 1.
3. 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. [22](#fig:022)).

Figure 22: Создание файла

Figure 22: Создание файла

Написала текст программы для вычисления выражения под вариантом 18, так как он попался мне по номеру моего студенческого билета (рис. [23](#fig:023)).

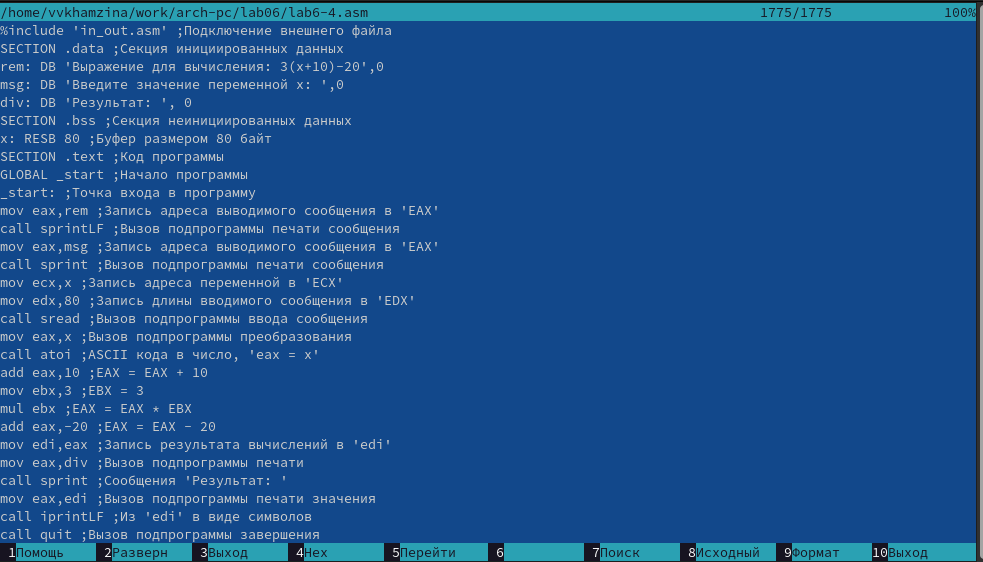


Figure 23: Редактирование файла

Код программы для вычисления выражения 18. 3(х+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 неинициированных 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 sprintLF ;Вызов подпрограммы печати сообщения  
mov eax,msg ;Запись адреса выводимого сообщения в 'EAX'  
call sprint ;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. [24](#fig:024)).

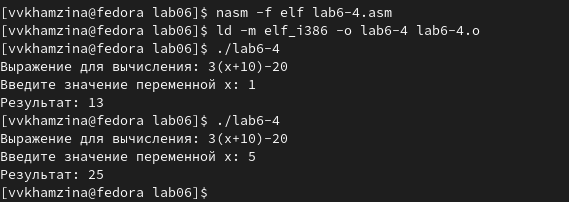


Figure 24: Запуск исполняемого файла

# 3 Выводы

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