Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Козина Дарья Александровна

Содержание

[Цель работы 2](#_Toc182044292)

[Задание 2](#_Toc182044293)

[Выполнение лабораторной работы 2](#_Toc182044294)

[Выполнение заданий для самостоятельной работы 6](#_Toc182044295)

[Выводы 9](#_Toc182044296)

Список иллюстраций

[Команда mc 2](#_Toc182044297)

[Открытый mc 2](#_Toc182044298)

[Перемещение между директориями 3](#_Toc182044299)

[Созданный каталог lab5. Создание файла lab5-1.asm 3](#_Toc182044300)

[Редактор программы 3](#_Toc182044301)

[Файл с программой 4](#_Toc182044302)

[Трансляция файла в объектный. Выполнение компоновки. Запуск полученного файла 4](#_Toc182044303)

[Установка файла 4](#_Toc182044304)

[Директории 4](#_Toc182044305)

[Копирование файла 5](#_Toc182044306)

[Копирование файла 5](#_Toc182044307)

[Редактирование файла 5](#_Toc182044308)

[Исполнение файла 5](#_Toc182044309)

[Отредактированный файл 6](#_Toc182044310)

[Исполнение файла 6](#_Toc182044311)

[Копирование файла 6](#_Toc182044312)

[Редактирование файла 7](#_Toc182044313)

[Исполнение файла 7](#_Toc182044314)

[Копирование файла 8](#_Toc182044315)

[Редактирование файла 8](#_Toc182044316)

[Исполнение файла 9](#_Toc182044317)

# Цель работы

Приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера mov и int.

# Задание

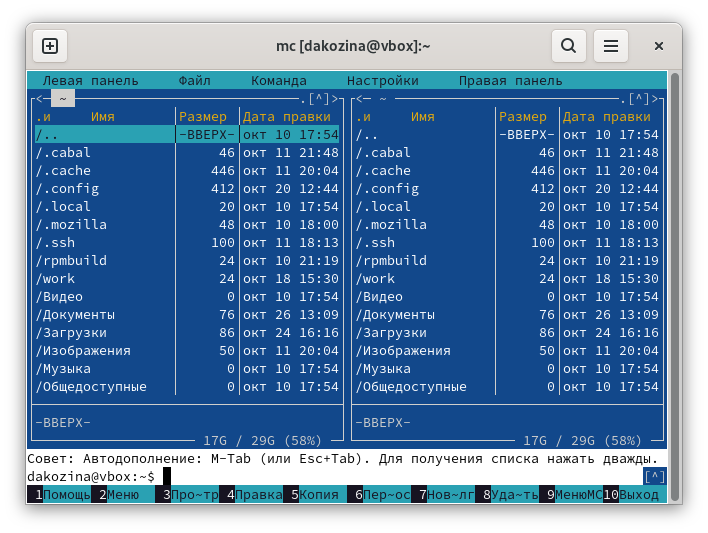
1. Основы работы с mc;
2. Структура программы на языке ассемблера NASM;
3. Подключение внешнего файла;
4. Выполнение заданий для самостоятельной работы.

# Выполнение лабораторной работы

С помощью команды mc открываем Midnight Commander (рис. [-@fig:001], [-@fig:002]).

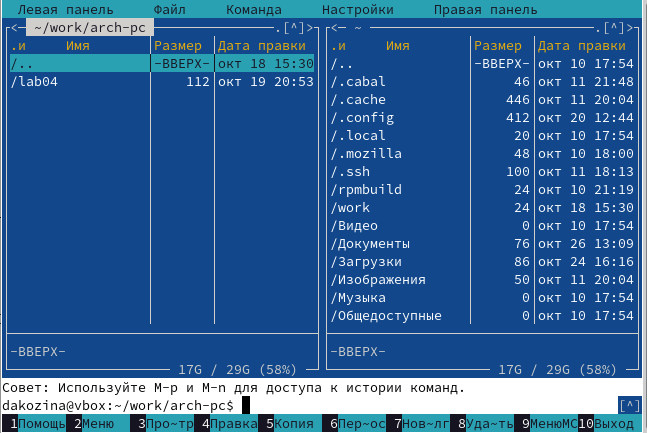
Команда mc

Команда mc



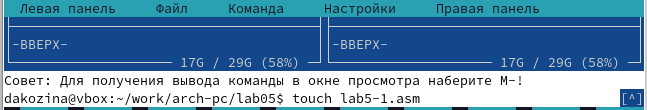
Открытый mc

С помощью клавиш переходим в каталог ~/work/arch-pc (рис. [-@fig:003]).



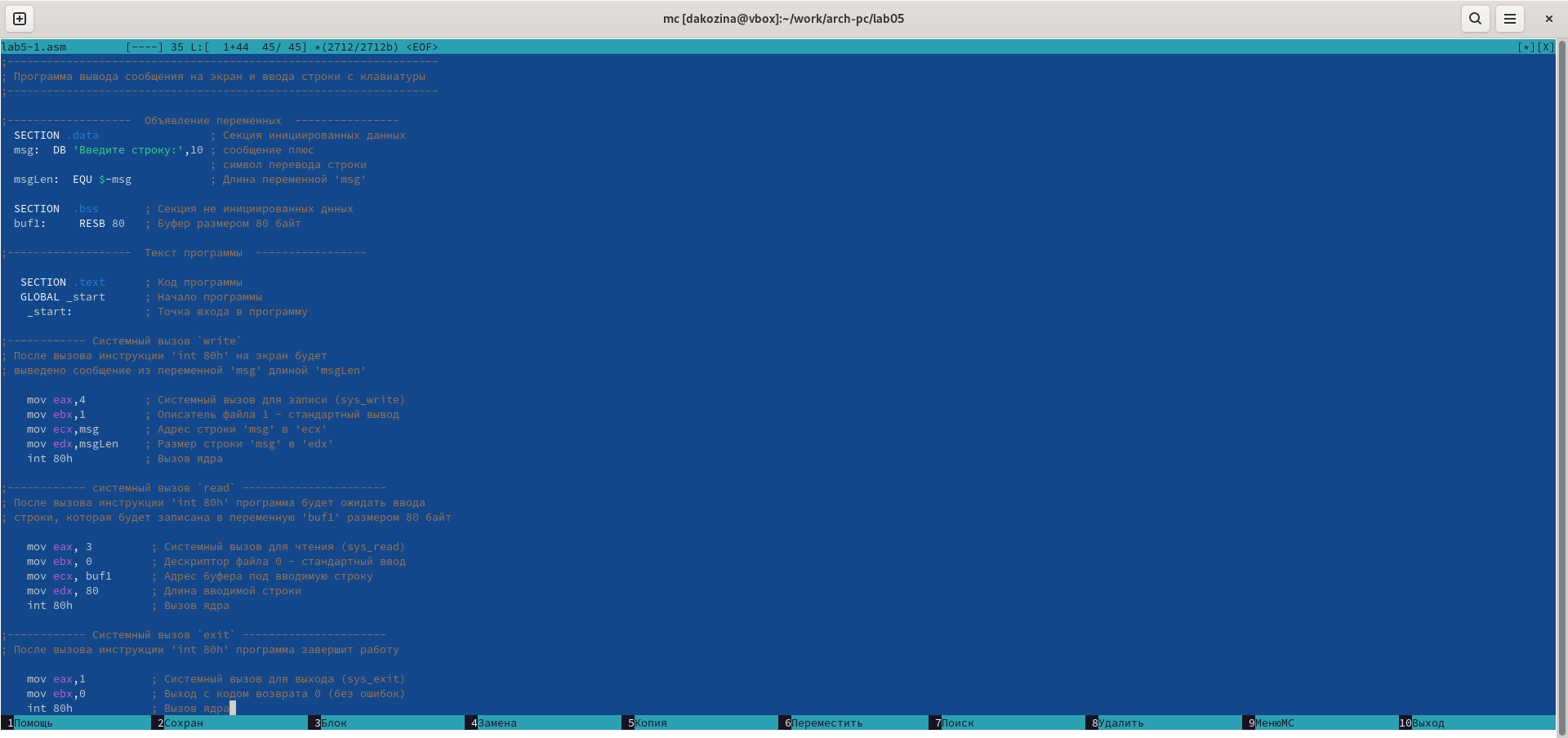
Перемещение между директориями

С помощью функциональной клавиши F7 создадим папку lab5 и перейдем в созданный каталог. Пользуясь строкой ввода, с помощью команды touch создадим файл lab5-1.asm (рис. [-@fig:004]).



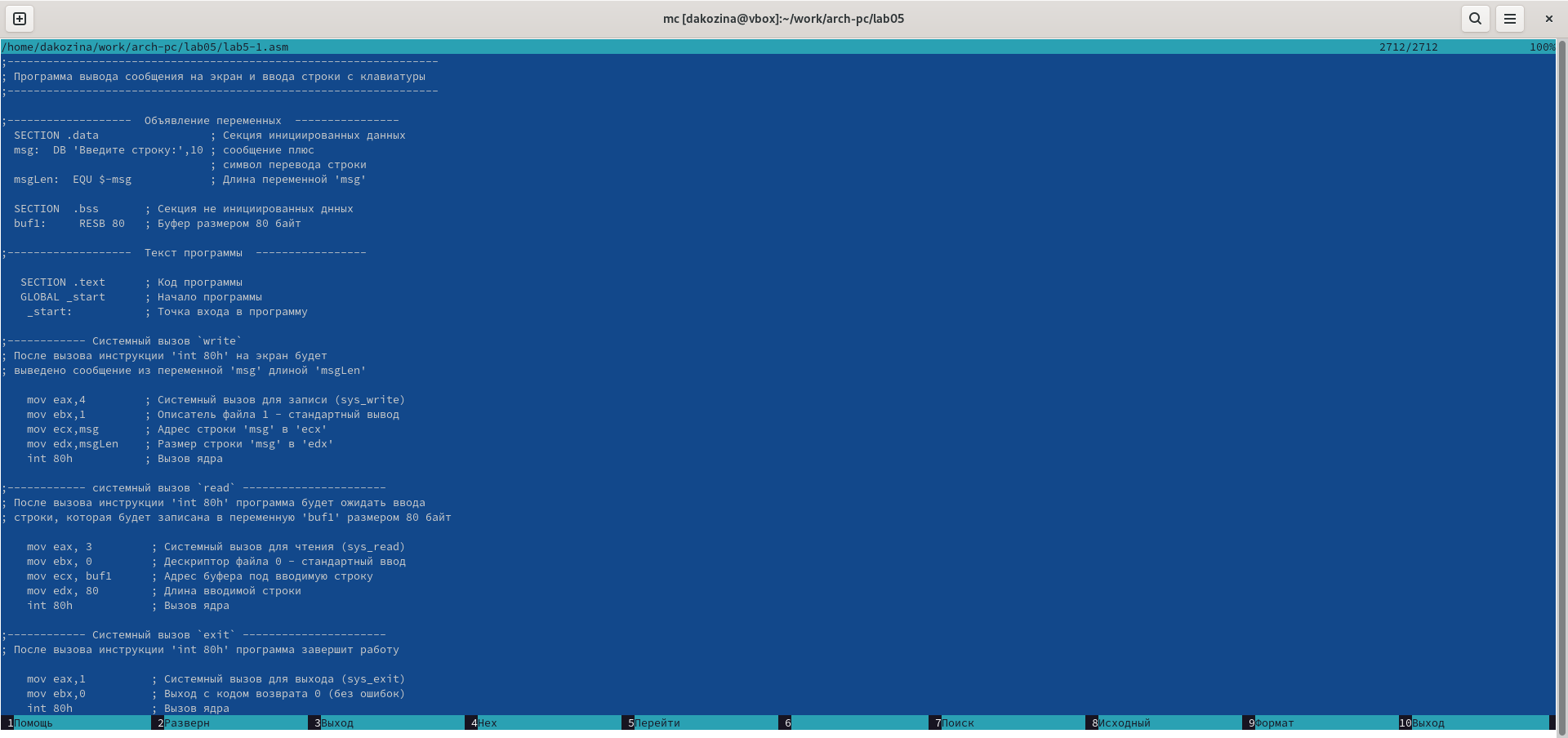
Созданный каталог lab5. Создание файла lab5-1.asm

С помощью функциональной клавиши F4 откроем созданный файл для редактирования во встроенном редакторе mcedit. Введем текст программы, сохраним изменения и закроем файл (рис. [-@fig:005]).



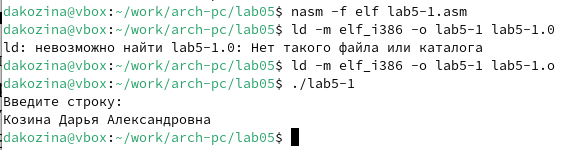
Редактор программы

С помощью функциональной клавиши F3 откроем файл для просмотра и убедимся, что файл содержит текст программы (рис. [-@fig:006]).



Файл с программой

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим полученный файл. На запрос ввести строку, введем ФИО (рис. [-@fig:007]).



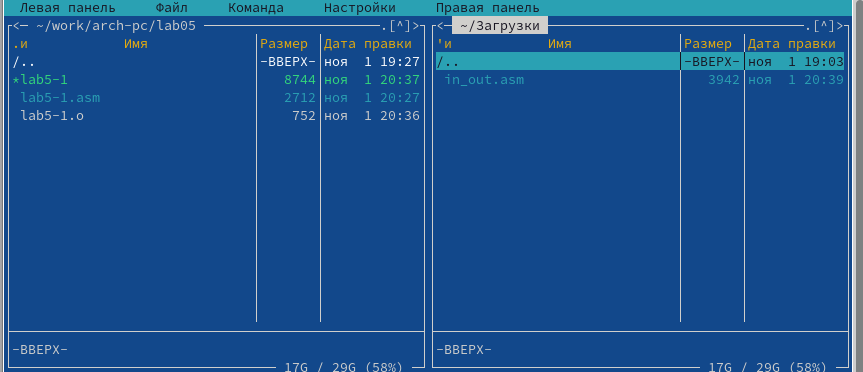
Трансляция файла в объектный. Выполнение компоновки. Запуск полученного файла

Скачаем файл in\_out.asm со страницы курса в ТУИС (рис. [-@fig:008]).



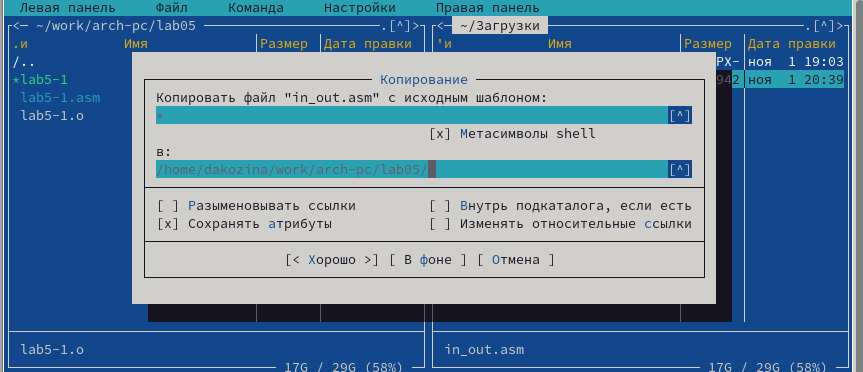
Установка файла

В одной панели mc откроем каталог с файлом lab5-1.asm, в другой панели каталог со скаченным файлом in\_out.asm (рис. [-@fig:009]).



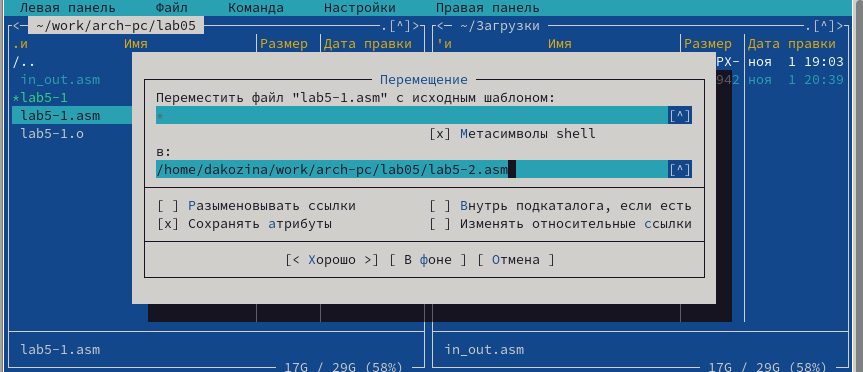
Директории

Скопируем файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5 (рис. [-@fig:010]).



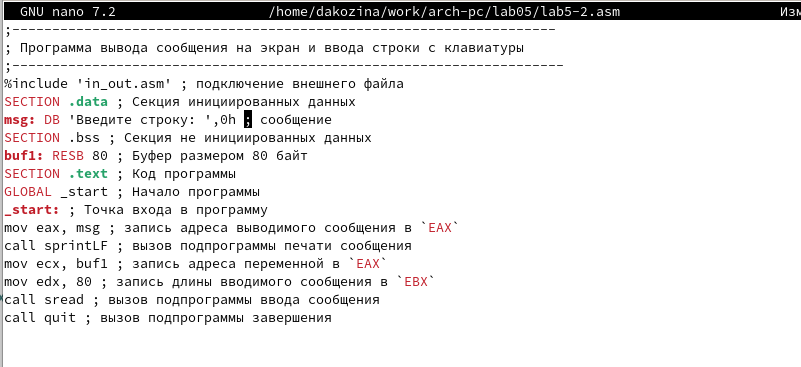
Копирование файла

С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm (рис. [-@fig:011]).



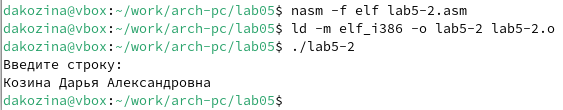
Копирование файла

Исправим текст программы в файле lab5-2.asm с использованием программы из внешнего файла in\_out.asm (рис. [-@fig:012]).



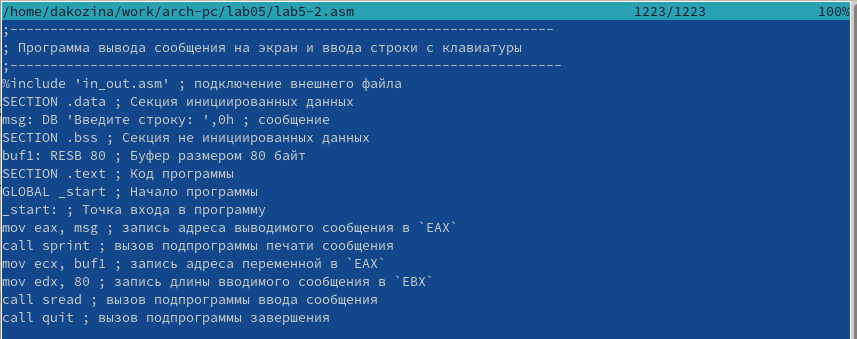
Редактирование файла

Создадим исполняемый файл и проверим его работу (рис. [-@fig:013]).



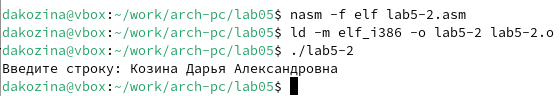
Исполнение файла

В файле lab5-2.asm заменим подпрограмму sprintLF на sprint (рис. [-@fig:014]).



Отредактированный файл

Создадим исполняемый файл и проверим его работу (рис. [-@fig:015]).

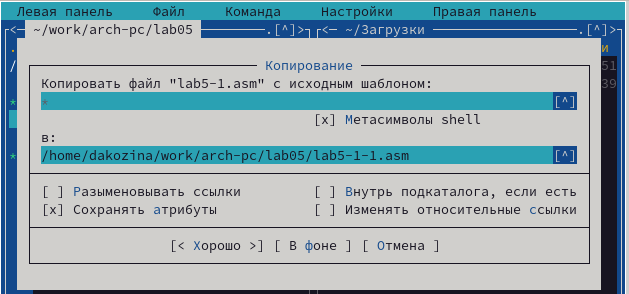


Исполнение файла

Разница между первым исполняемым файлом и вторым в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается разница между подпрограммами sprintLF и sprint.

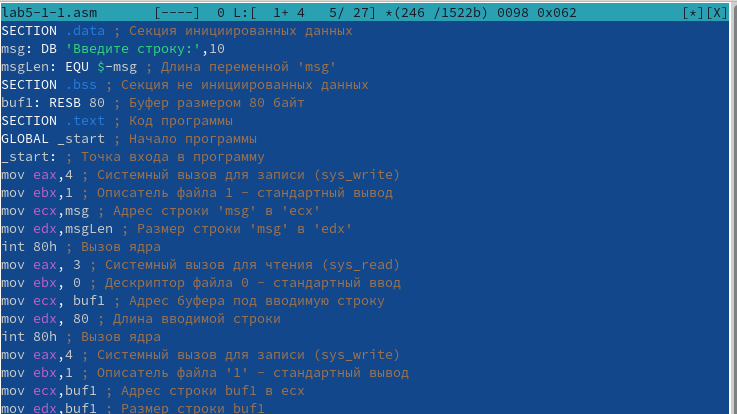
# Выполнение заданий для самостоятельной работы

С помощью функциональной клавиши F5 создадим копию файла lab5-1.asm с именем lab5-1-1.asm (рис. [-@fig:016]).



Копирование файла

С помощью функциональной клавиши F4 откроем созданный файл в редакторе и внесем изменения в программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую строку (рис. [-@fig:017]).



Редактирование файла

Код измененной программы:

SECTION .data ; Секция инициированных данных

msg: DB ‘Введите строку:’,10

msgLen: EQU $-msg ; Длина переменной ‘msg’

SECTION .bss ; Секция не инициированных данных

buf1: RESB 80 ; Буфер размером 80 байт

SECTION .text ; Код программы

GLOBAL \_start ; Начало программы

\_start: ; Точка входа в программу

mov eax,4 ; Системный вызов для записи (sys\_write)

mov ebx,1 ; Описатель файла 1 - стандартный вывод

mov ecx,msg ; Адрес строки ‘msg’ в ‘ecx’

mov edx,msgLen ; Размер строки ‘msg’ в ‘edx’

int 80h ; Вызов ядра

mov eax, 3 ; Системный вызов для чтения (sys\_read)

mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод

mov ecx, buf1 ; Адрес буфера под вводимую строку

mov edx, 80 ; Длина вводимой строки

int 80h ; Вызов ядра mov eax,4 ; Системный вызов для записи (sys\_write)

mov ebx,1 ; Описатель файла ‘1’ - стандартный вывод

mov ecx,buf1 ; Адрес строки buf1 в ecx

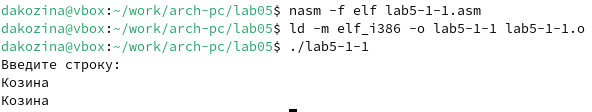
mov edx,buf1 ; Размер строки buf1

int 80h ; Вызов ядра mov eax,1 ; Системный вызов для выхода (sys\_exit)

mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)

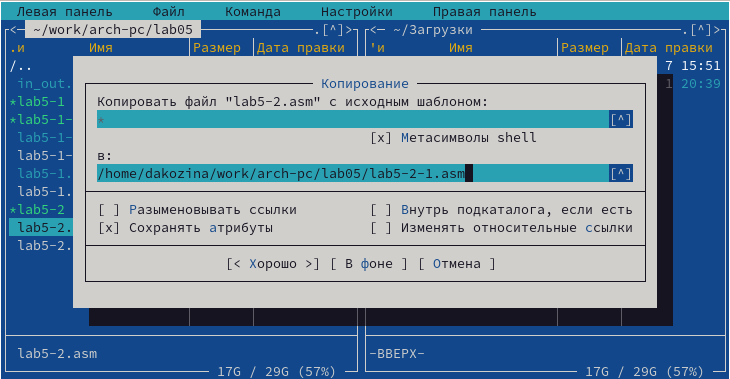
int 80h ; Вызов ядра

Проверим работу полученного файла(рис. [-@fig:018]).



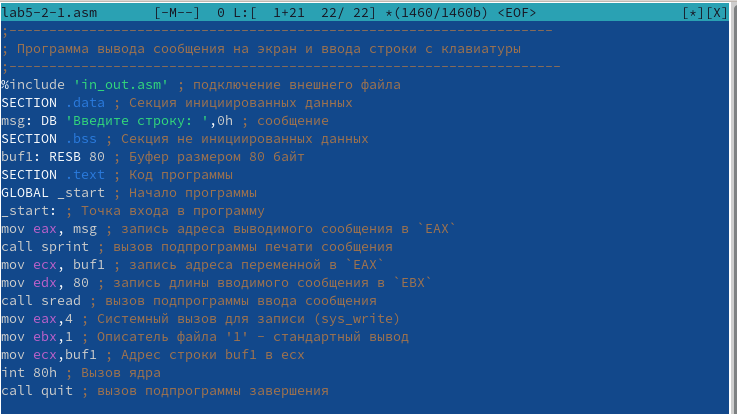
Исполнение файла

С помощью функциональной клавиши F5 создадим копию файла lab5-2.asm с именем lab5-2-1.asm (рис. [-@fig:019]).



Копирование файла

С помощью функциональной клавиши F4 откроем созданный файл в редакторе и внесем изменения в программу с использованием подпрограмм из внешнего файла in\_out.asm так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую строку (рис. [-@fig:020]).



Редактирование файла

Код измененной программы:

%include ‘in\_out.asm’

SECTION .data ; Секция инициированных данных

msg: DB ‘Введите строку:’,0h ; сообщение

SECTION .bss ; Секция не инициированных данных

buf1: 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 sprint ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; запись адреса переменной в EAX

mov edx, 80 ; запись длины вводимого сообщения в EBX

call sread ; вызов подпрограммы ввода сообщения

mov eax,4 ; Системный вызов для записи (sys\_write)

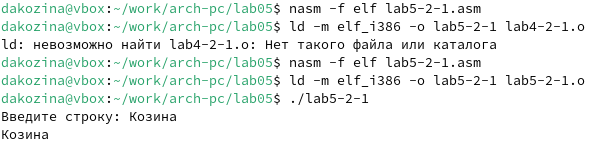
mov ebx,1 ; Описатель файла ‘1’ - стандартный вывод

mov ecx,buf1 ; Адрес строки buf1 в ecx

int 80h ; Вызов ядра

call quit ; вызов подпрограммы завершения

Проверим работу созданного файла (рис. [-@fig:021]).



Исполнение файла

# Выводы

В ходе лабораторной работы мы приобрели практические навыки работы в Midnight Commander, освоили инструкции языка ассемблера mov и int.