Лабораторная работа №6

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

Вершинина Ангелина Алексеевна

Содержание

# 1 Цель работы

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

# 2 Задание

Написать программы принимающие на ввод строку и вывод ее.

# 3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управ- лению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной.

# 4 Выполнение лабораторной работы

Открою Midnight Commander (рис. 1)

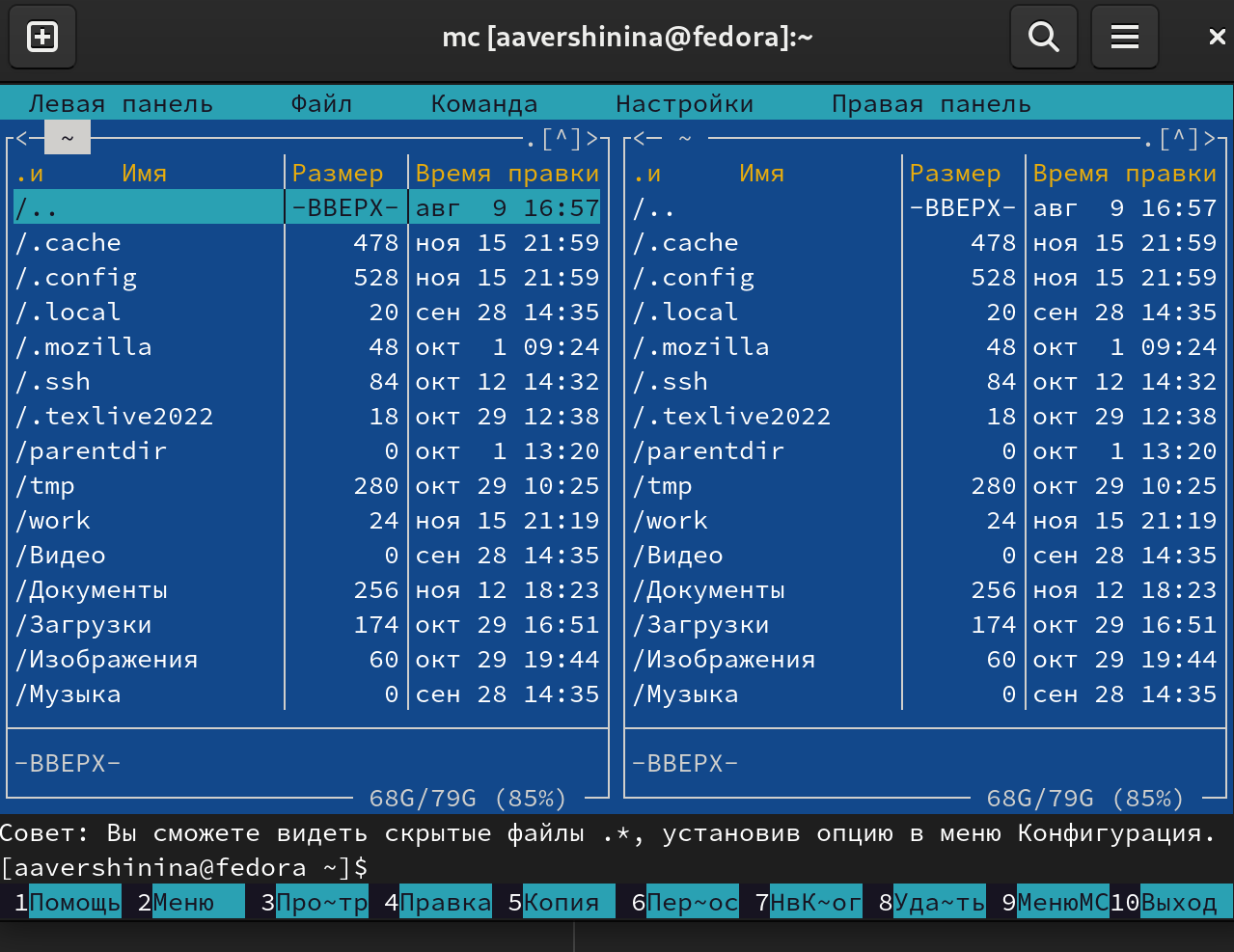


Рис. 1: Открытие Midnight Commander

Пользуясь клавишами ↑ , ↓ и Enter перейду в каталог ~/work/arch-pc (рис. 2)

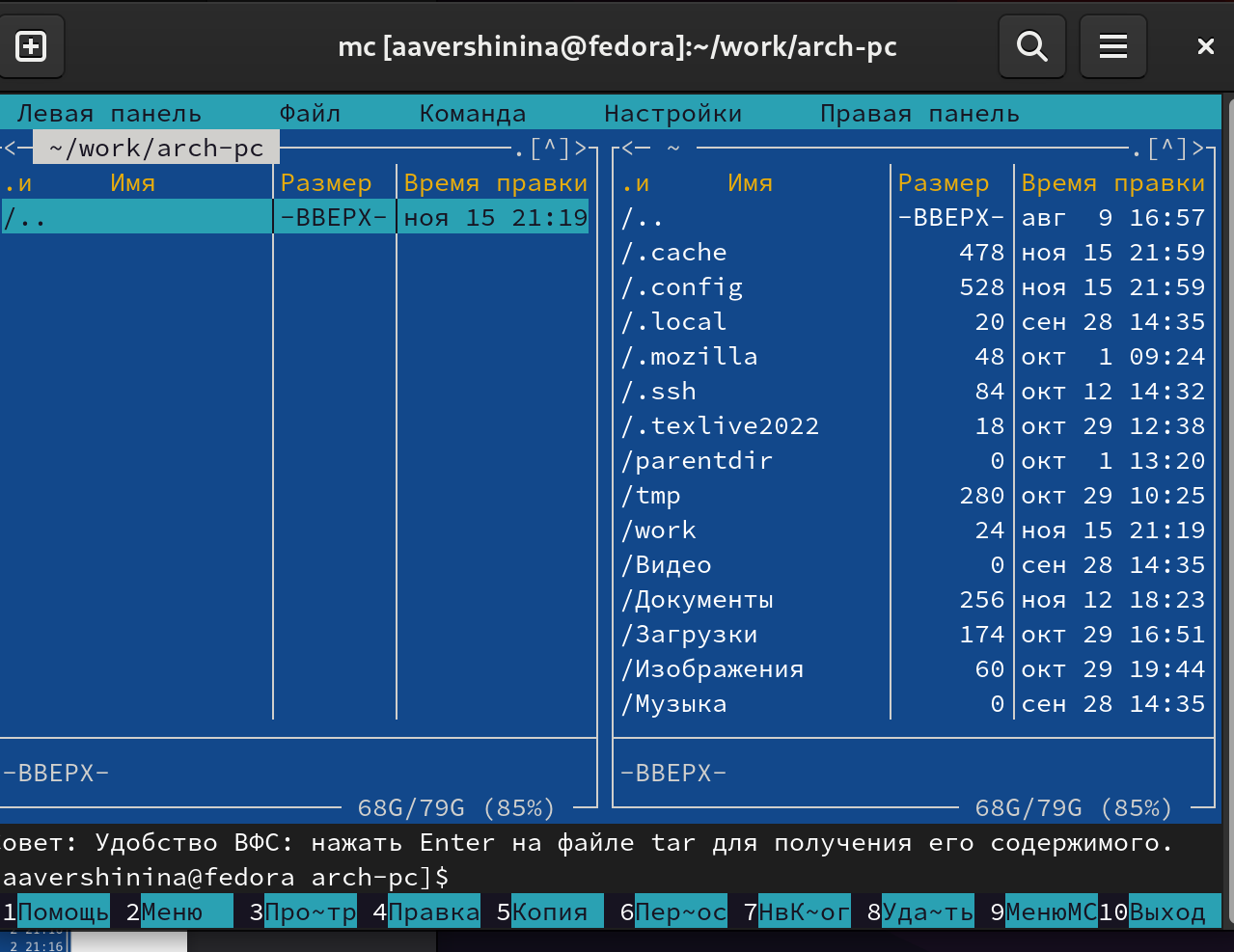


Рис. 2: Переход в каталог

С помощью функциональной клавиши F7 создам папку lab06 (рис. 3) и перейду в созданный каталог.

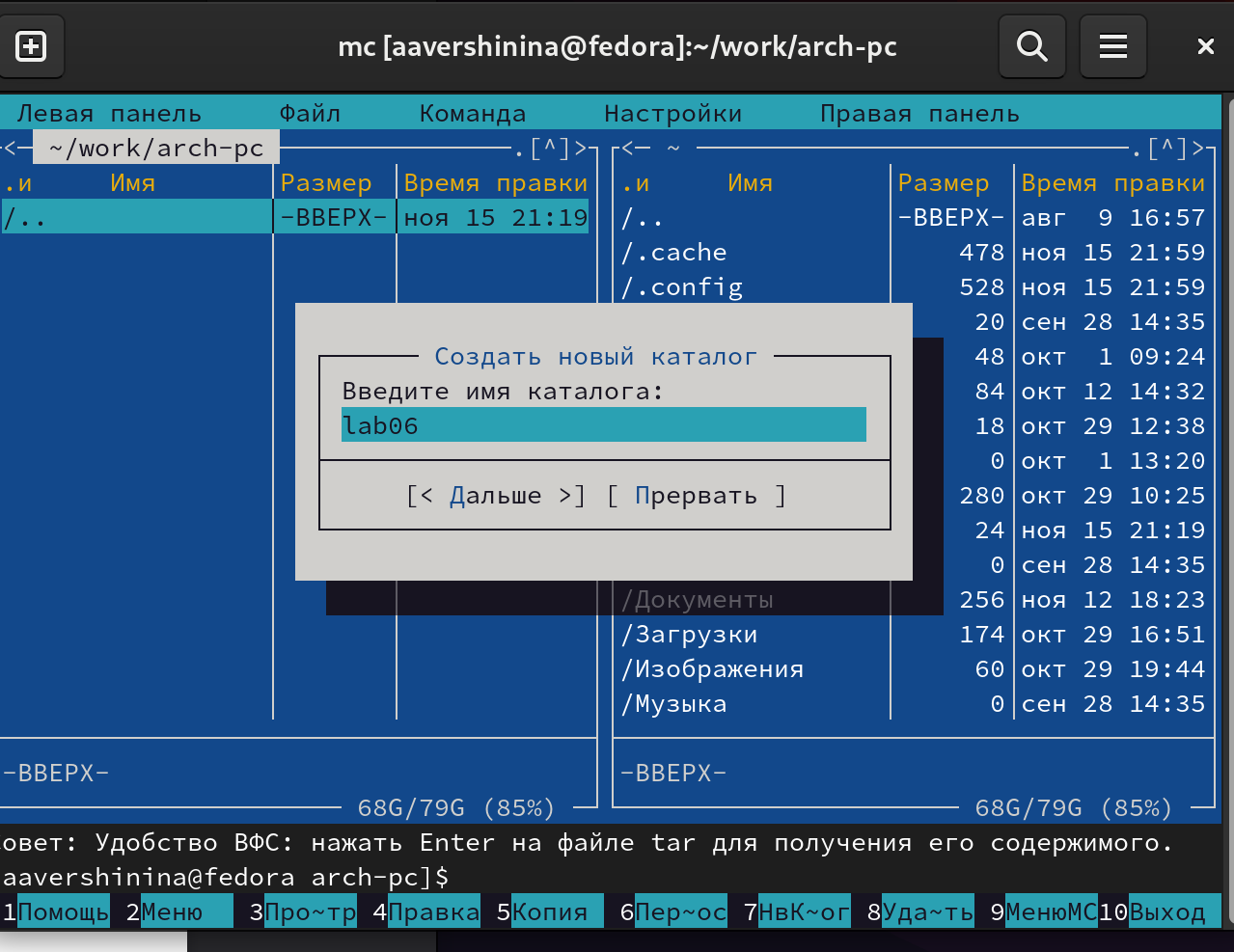


Рис. 3: Создание каталога

Пользуясь строкой ввода и командой touch создам файл lab6-1.asm (рис. 4)

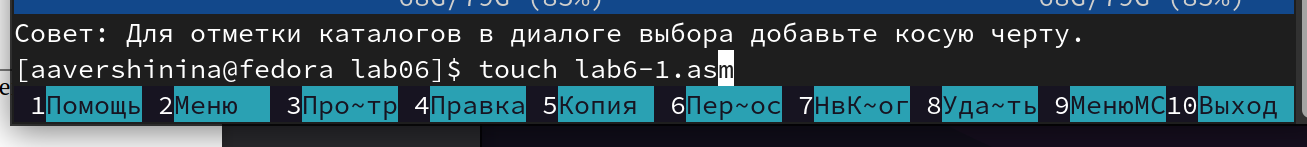


Рис. 4: Создание файла

С помощью функциональной клавиши F4 открою файл lab6-1.asm для редактирования во встроенном редакторе (рис. 5)



Рис. 5: Открытие файла

Введу текст программы из листинга 6.1 (рис. 6), со- храню изменения (рис. 7) и закрою файл.

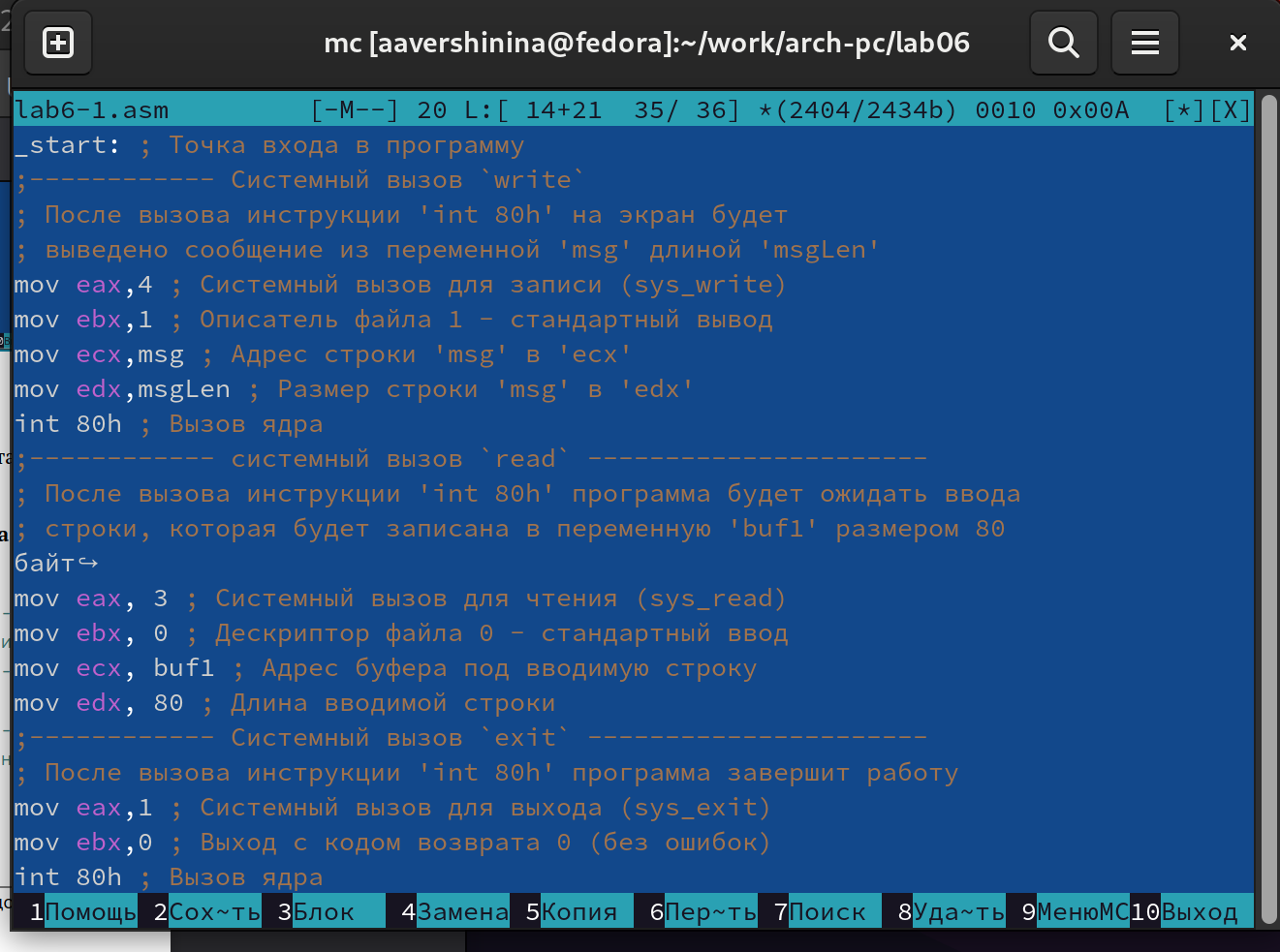


Рис. 6: Ввод текста программы

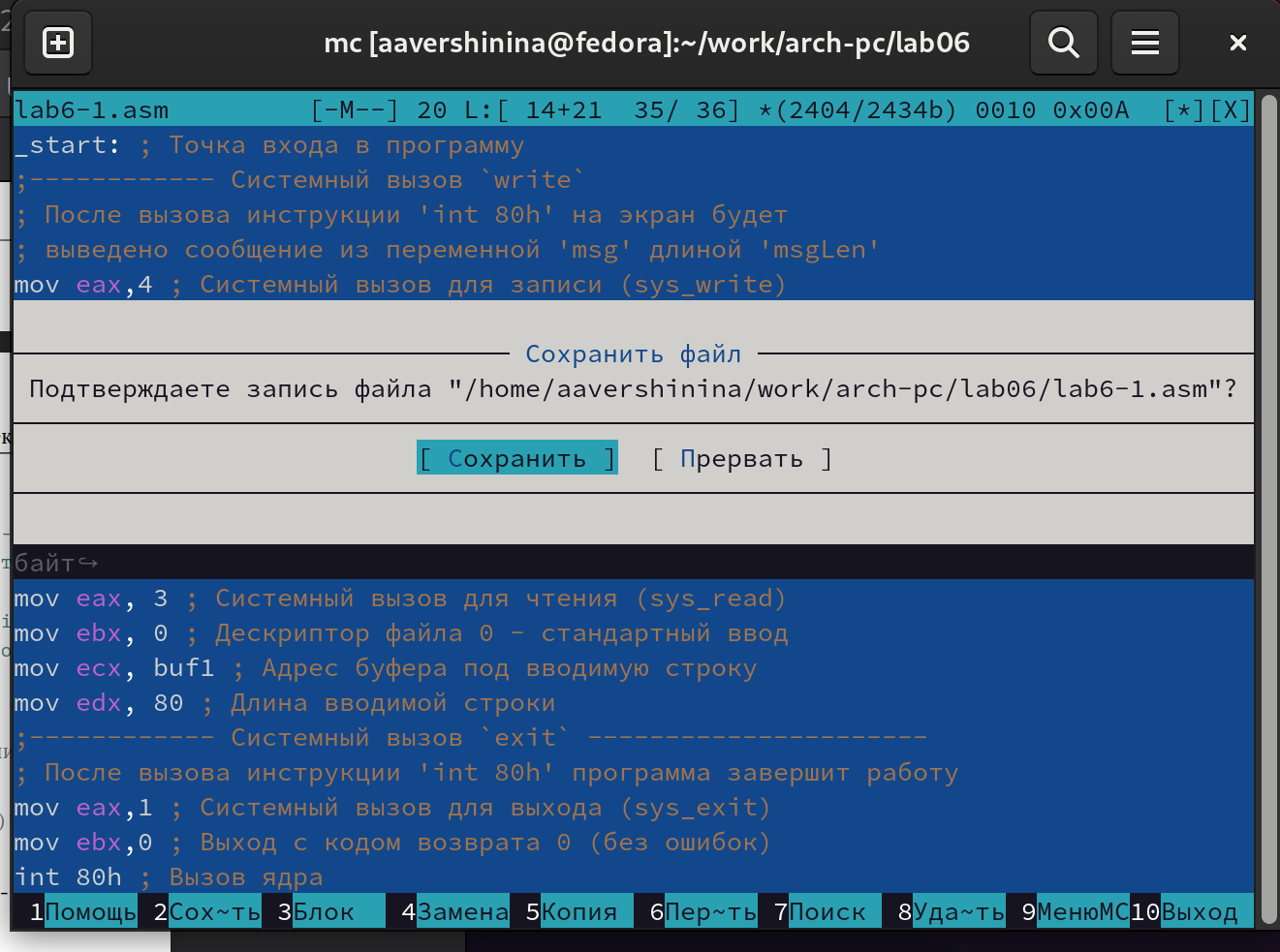


Рис. 7: Сохранение текста программы

С помощью функциональной клавиши F3 открою файл lab6-1.asm для просмотра. Файл действительно содержит текст программы.

Оттранслирую текст программы lab6-1.asm в объектный файл. Выполню компоновку объектного файла и запущу получившийся исполняемый файл. Программа выводит строку ‘Введите строку:’ и ожидает ввода с клавиатуры. На запрос введу фамилию и имя (рис. 8)

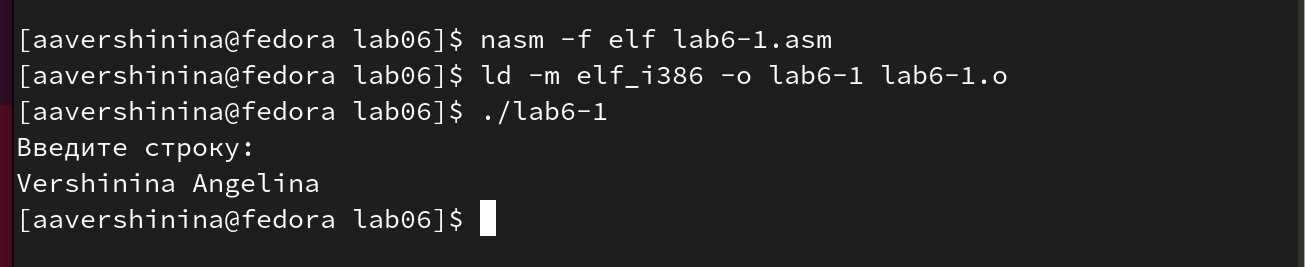


Рис. 8: Запуск исполняемого файла

## 4.1 Подключение внешнего файла in\_out.asm

Скачаю файл in\_out.asm со страницы курса в ТУИС. В одной из панелей mc открою каталог с файлом lab6-1.asm. В другой панели каталог со скаченным файлом in\_out.asm. (рис. 9)

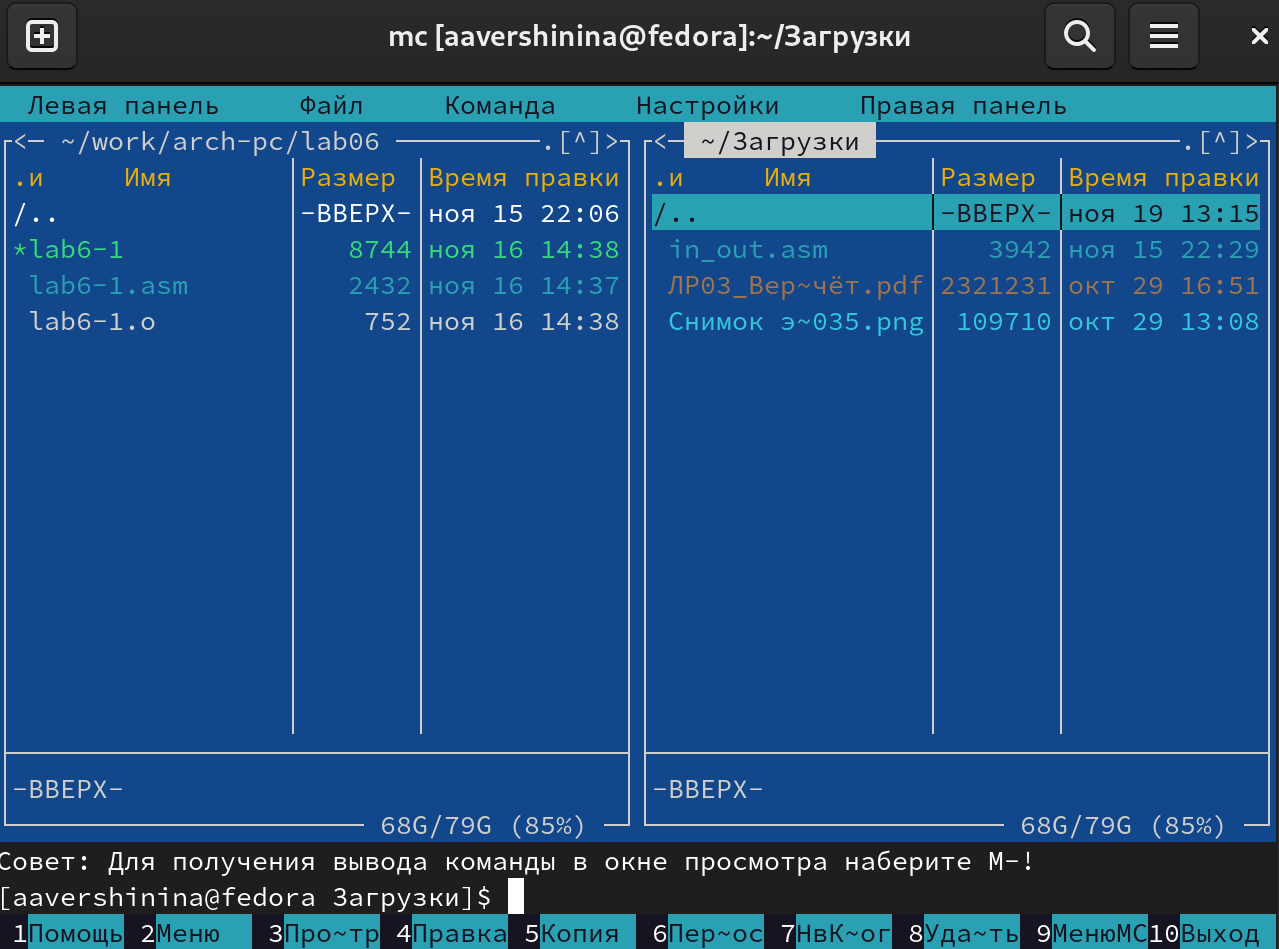


Рис. 9: Открытие нужных файлов в панели

Скопирую файл in\_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши F5 (рис. 10)

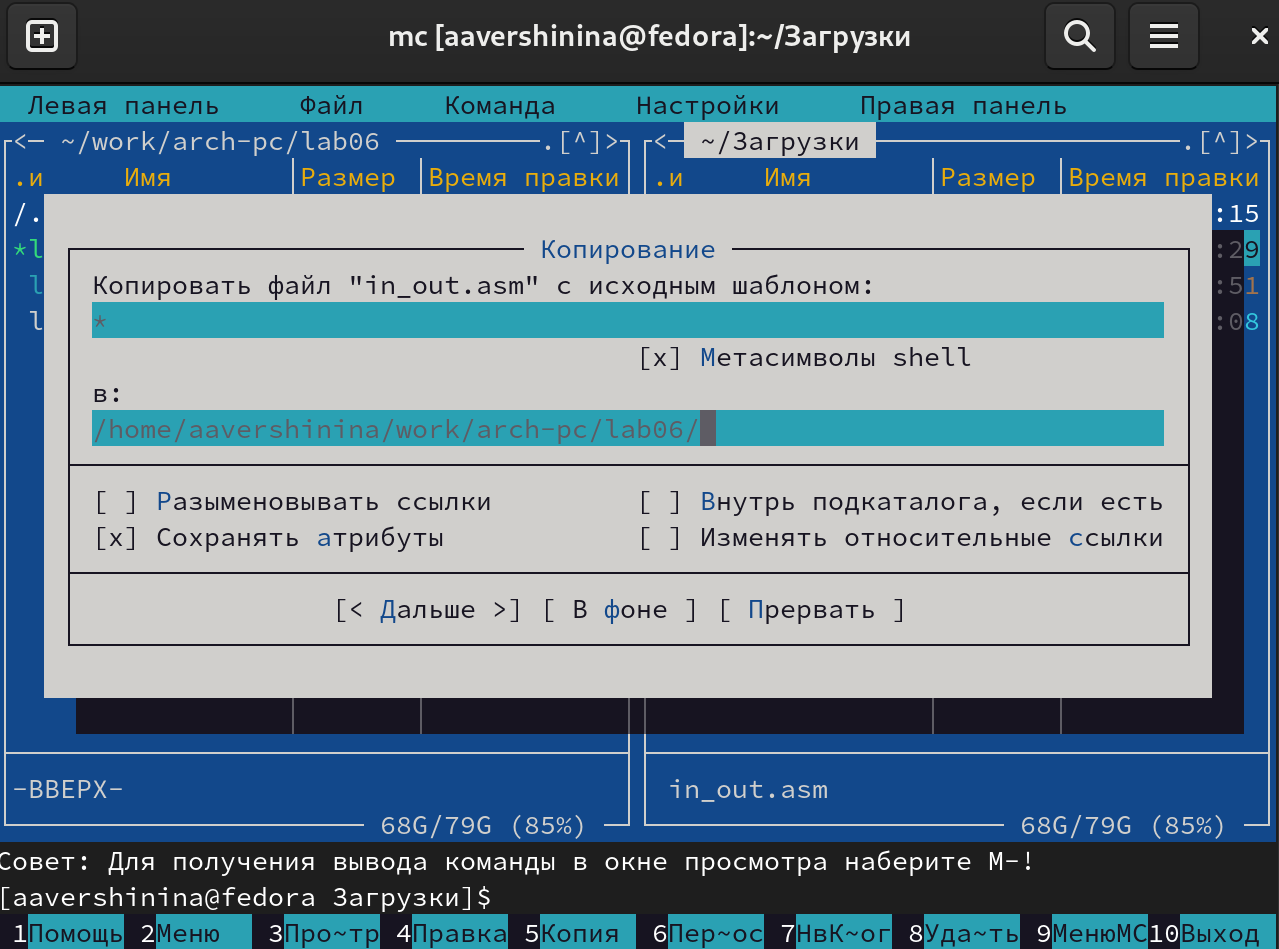


Рис. 10: Копирование файла

С помощью функциональной клавиши F6 создам копию файла lab6- 1.asm с именем lab6-2.asm. Выделю файл lab6-1.asm, нажму клавишу F6 , введу имя файла lab6-2.asm и нажму клавишу Enter (рис. 11)

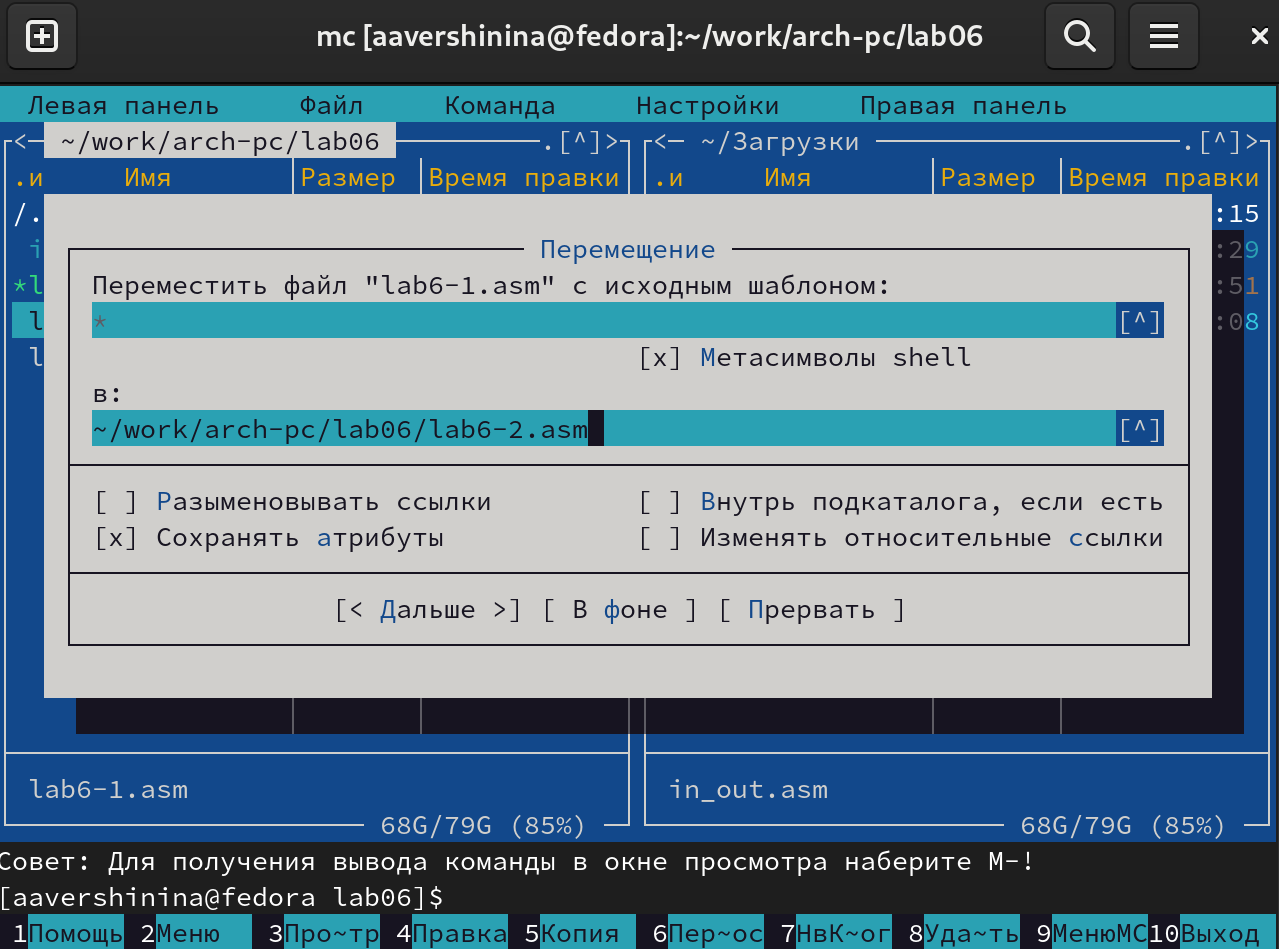


Рис. 11: Открытие файла

Исправлю текст программы в файле lab6-2.asm с использование под- программ из внешнего файла in\_out.asm в соответствии с листингом 6.2. Создам испол- няемый файл и проверьте его работу (рис. 12)

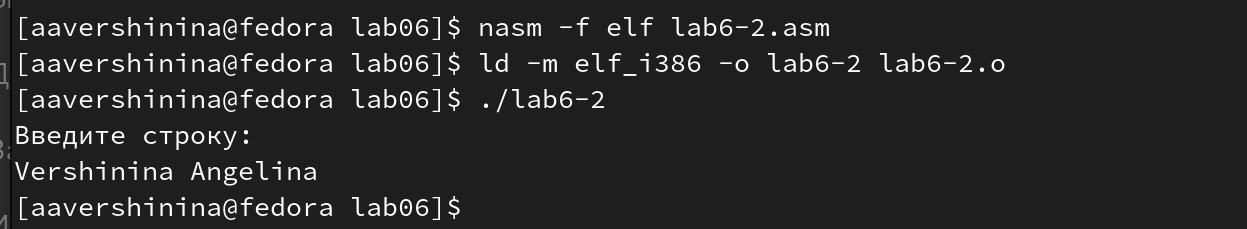


Рис. 12: Запуск исполняемого файла

В файле lab6-2.asm заменю подпрограмму sprintLF на sprint. (рис. 13) Создам исполняемый файл и проверю его работу.(рис. 14) Отличие sprintLF и sprint заключается в том, что в первом случае программа запрашивает ввод данных с новой строки, а во втором не происходит переноса строки для ввода данных.

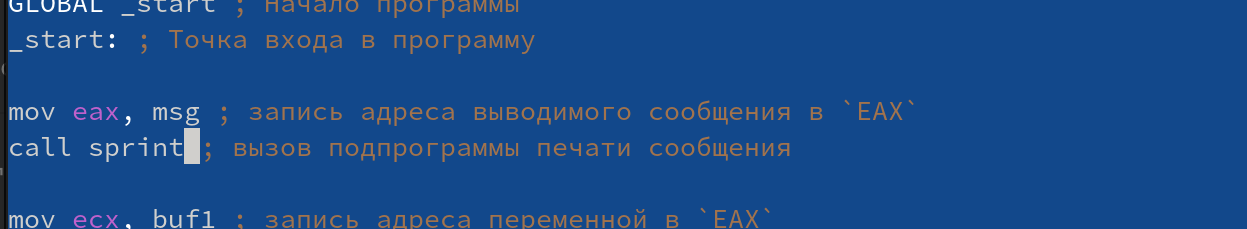


Рис. 13: Редактирование файла

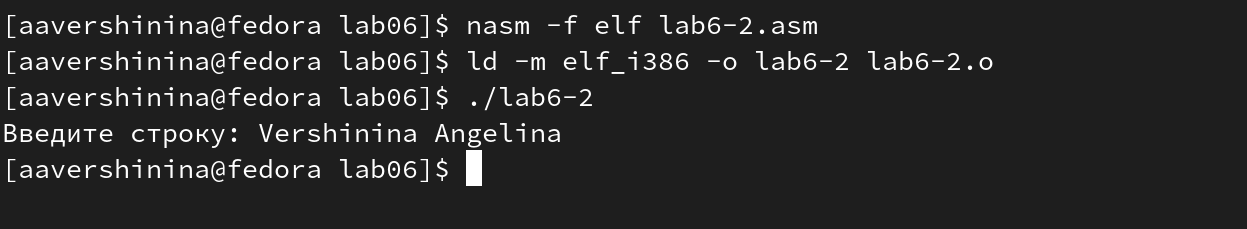


Рис. 14: Запуск исполняемого файла

## 4.2 Выполнение заданий для самостоятельной работы

1. Создам копию файла lab6-1.asm с именем lab6-11.asm с помощью функциональной клавиши F5. С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменю программу так, чтобы кроме вывода приглашения и запроса ввода, производился вывод вводимой пользователем строки (рис. 15)

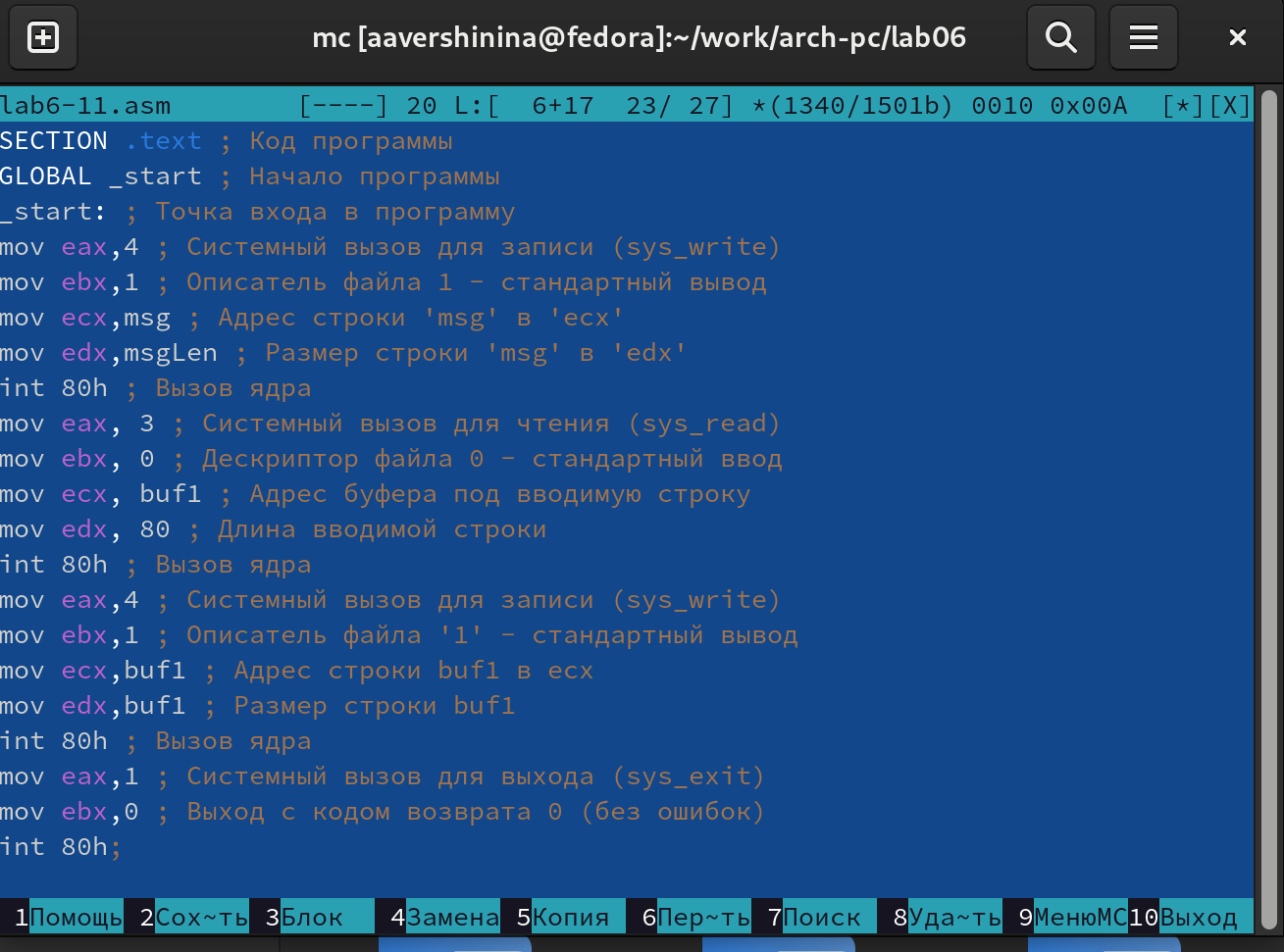


Рис. 15: Редактирование файла

Создам исполняемый файл и проверю его работу. Программа запрашивает ввод строки, ввожу свою фамилию, программа выводит введенную строку(рис. 16)

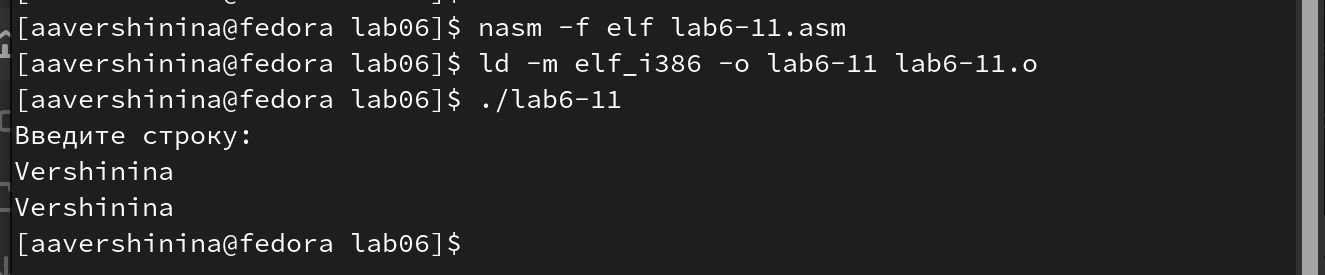


Рис. 16: Запуск исполняемого файла

**Листинг программы 1**

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

1. Создам копию файла lab6-2.asm с именем lab6-2-1.asm с помощью функциональной клавиши F5. С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменю программу так, чтобы кроме вывода приглашения и запроса ввода, производился вывод вводимой пользователем строки.

Создам исполняемый файл и проверю его работу. Программа запрашивает ввод строки, ввожу свою фамилию, программа выводит введенную строку(рис. 17)

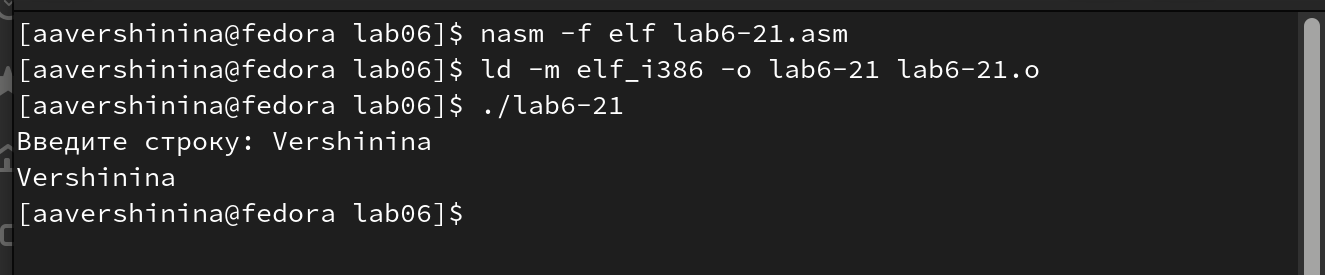


Рис. 17: Запуск исполняемого файла

**Листинг программы 2**

%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

# 5 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера mov и int.

# Список литературы