Отчет по лабораторной работе №5

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

Мальянц В. К.

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Открываю Midnight Commander с помощью команды mc (рис. 1).

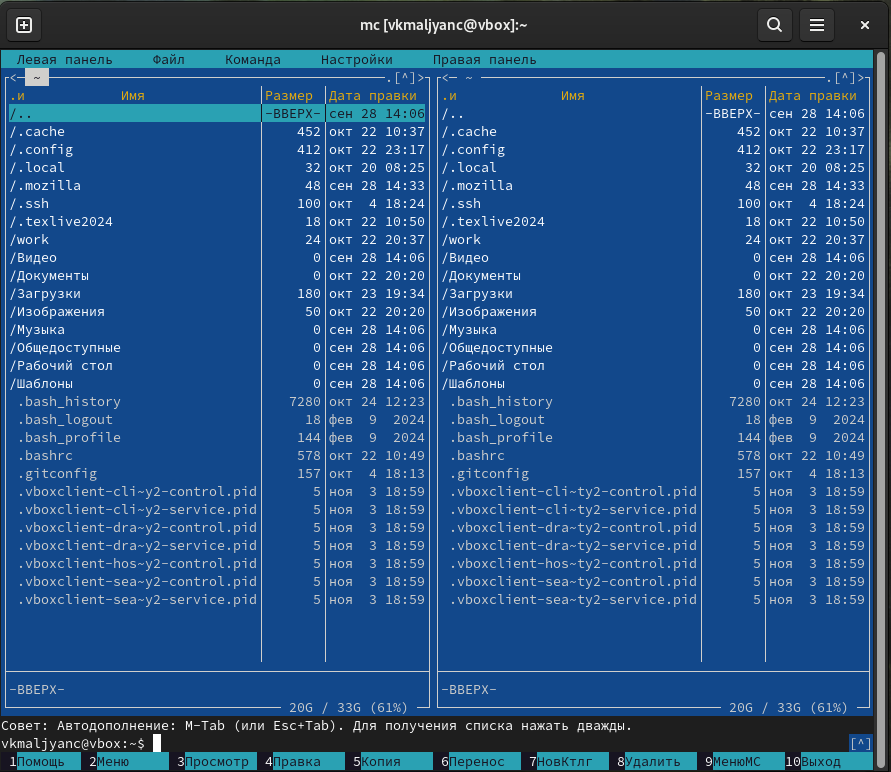


Рис. 1: Окно mc

Перехожу в каталог ~/work/arch-pc с помощью файлового менеджера mc (рис. 2).

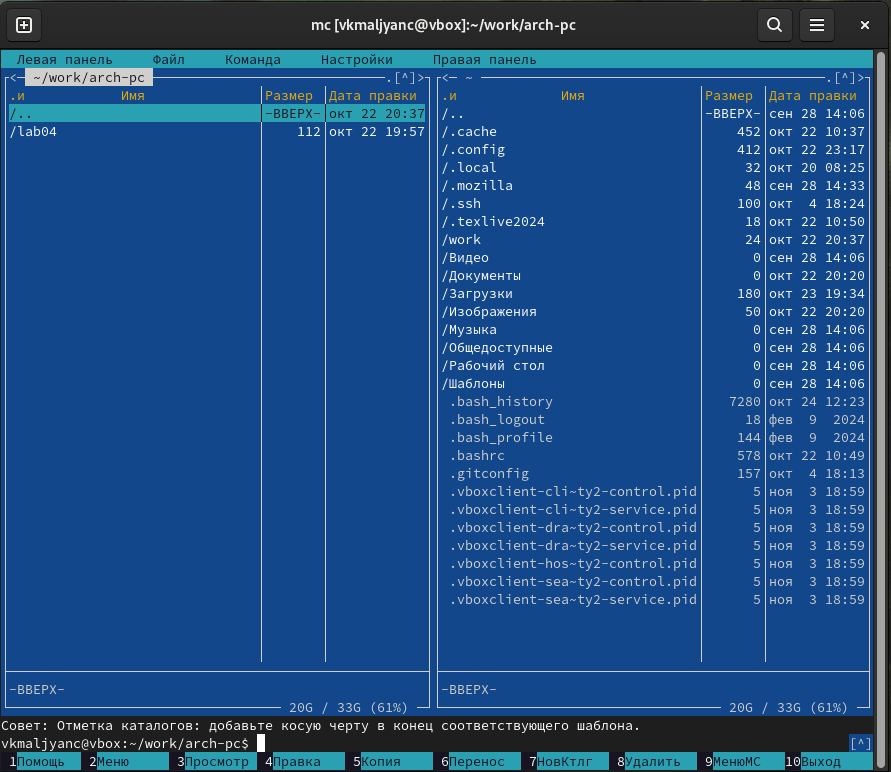


Рис. 2: Перемещение в каталог ~/work/arch-pc

Создаю папку lab05 с помощью функциональной клавиши F7 (рис. 3) и перехожу в нее (рис. 4).

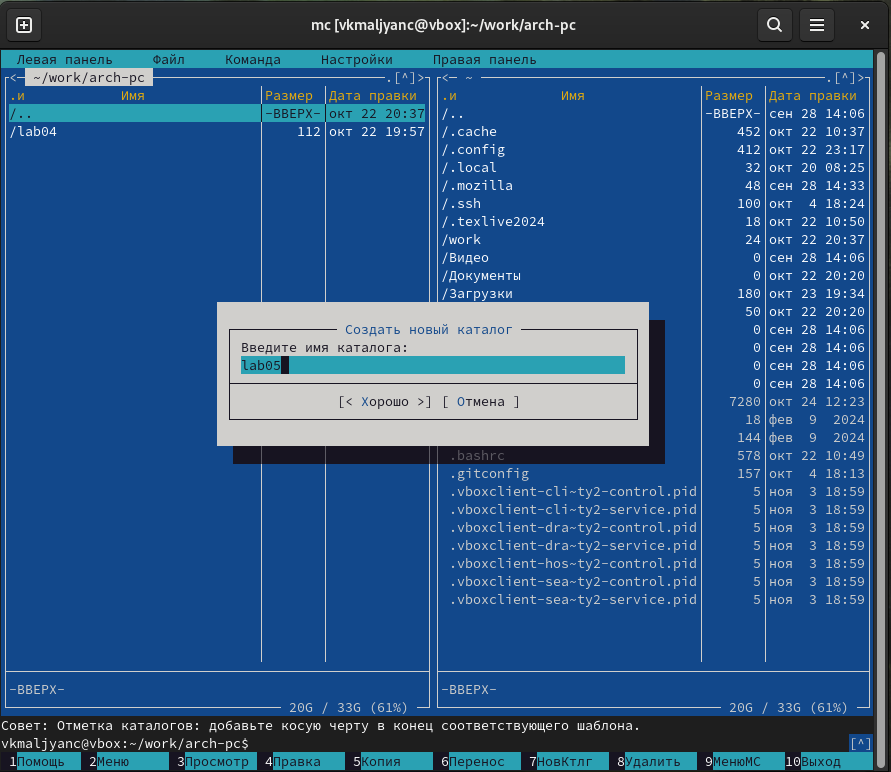


Рис. 3: Создание папки lab05

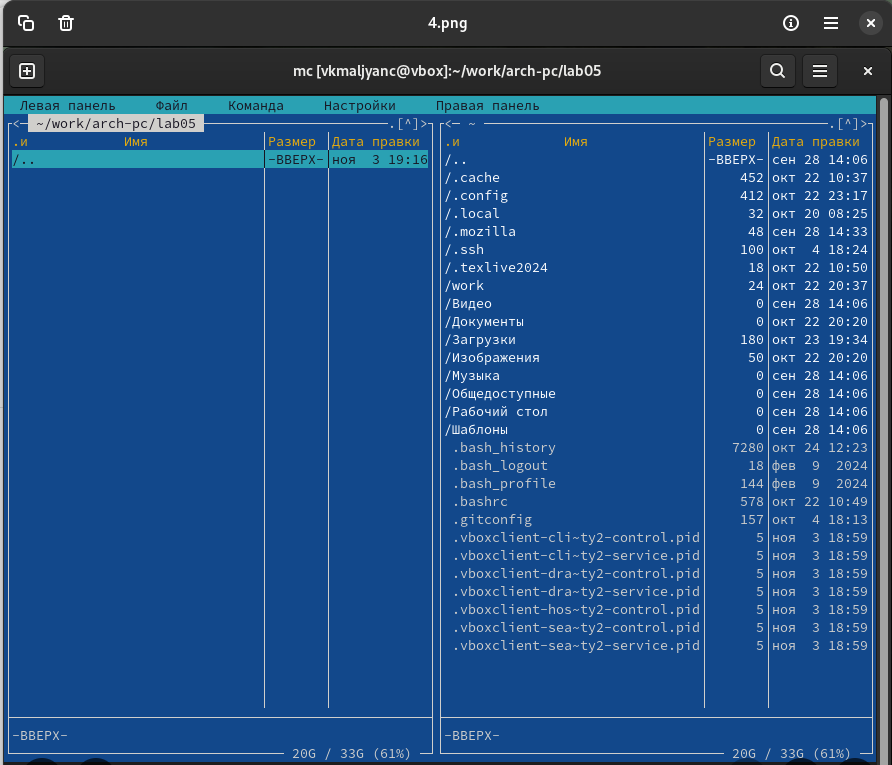


Рис. 4: Переход в папку lab05

В строке ввода с помощью команды touch создаю файл lab-1.asm (рис. 5).

Рис. 5: Создание файла lab-1.asm

Рис. 5: Создание файла lab-1.asm

Открываю файл lab5-1.asm для редактирования во встроенном редакторе с помощью функциональной клавиши F4 (рис. 6).

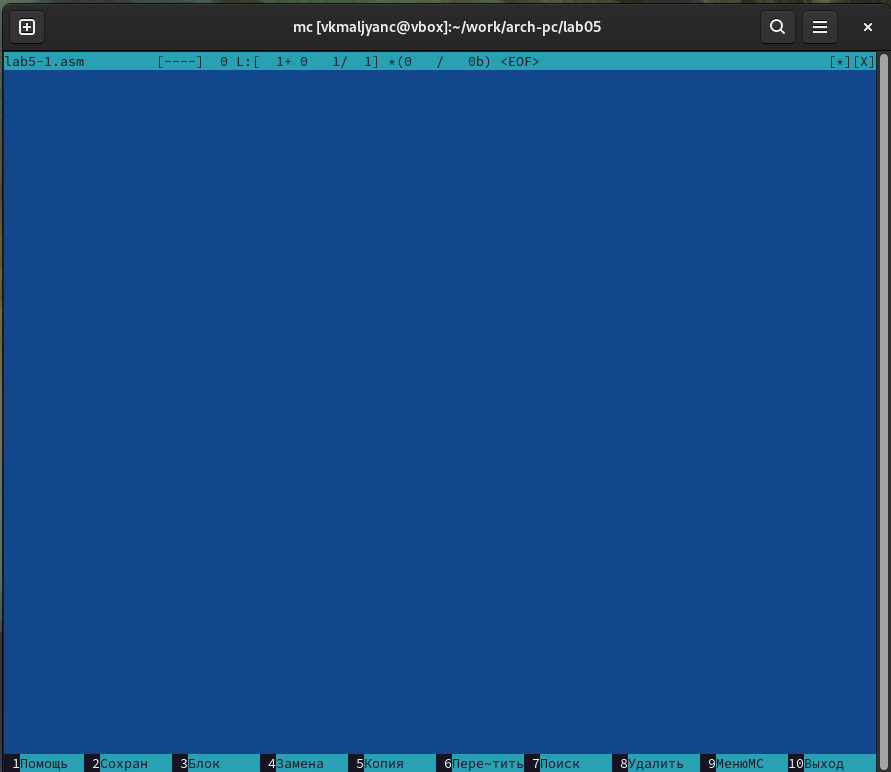


Рис. 6: Открытие файла lab5-1.asm для редактирования во встроенном редакторе

Ввожу текст программы вывода сообщения на экран и ввода строки с клавиатуры (рис. 7).

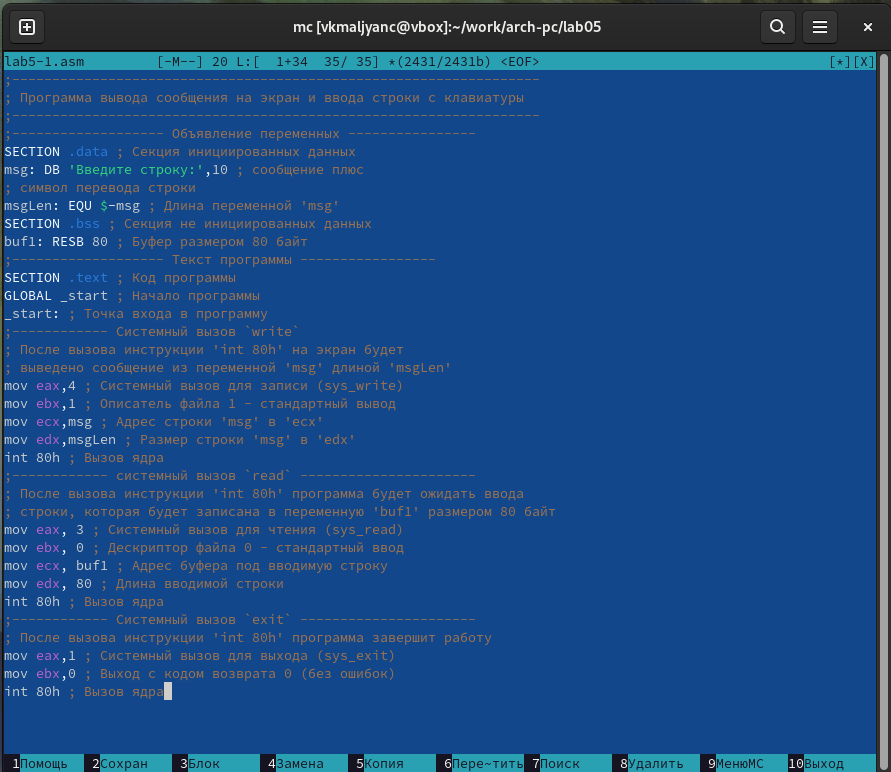


Рис. 7: Ввод текста программы

Сохраняю изменения с помощью функциональной клавиши F2 (рис. 8).

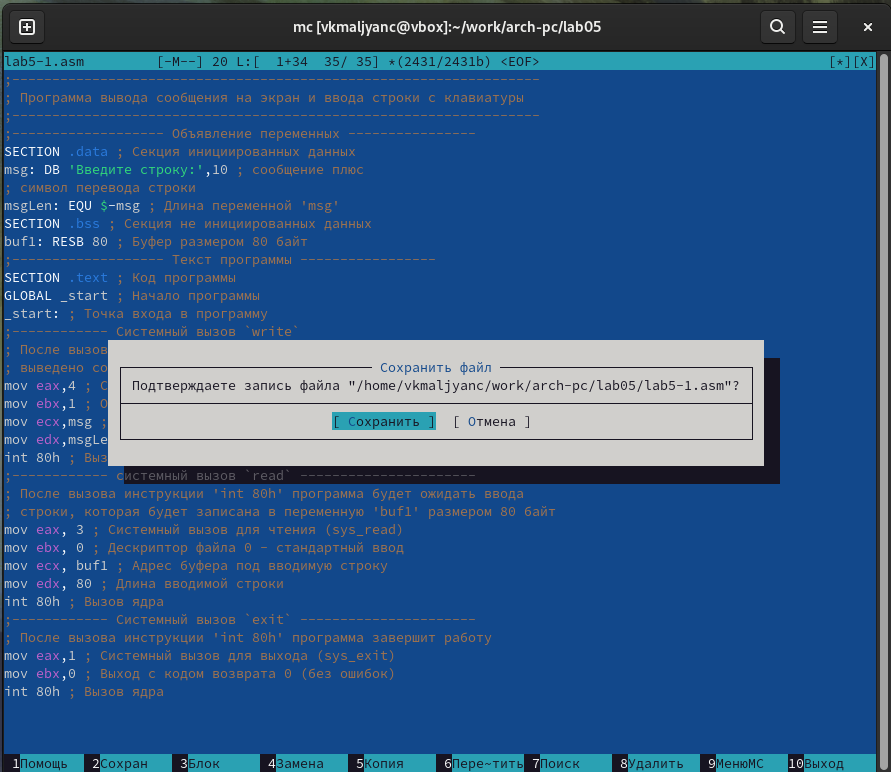


Рис. 8: Сохранение изменений

Выхожу из редактора с помощью функциональной клавиши F10 (рис. 9).

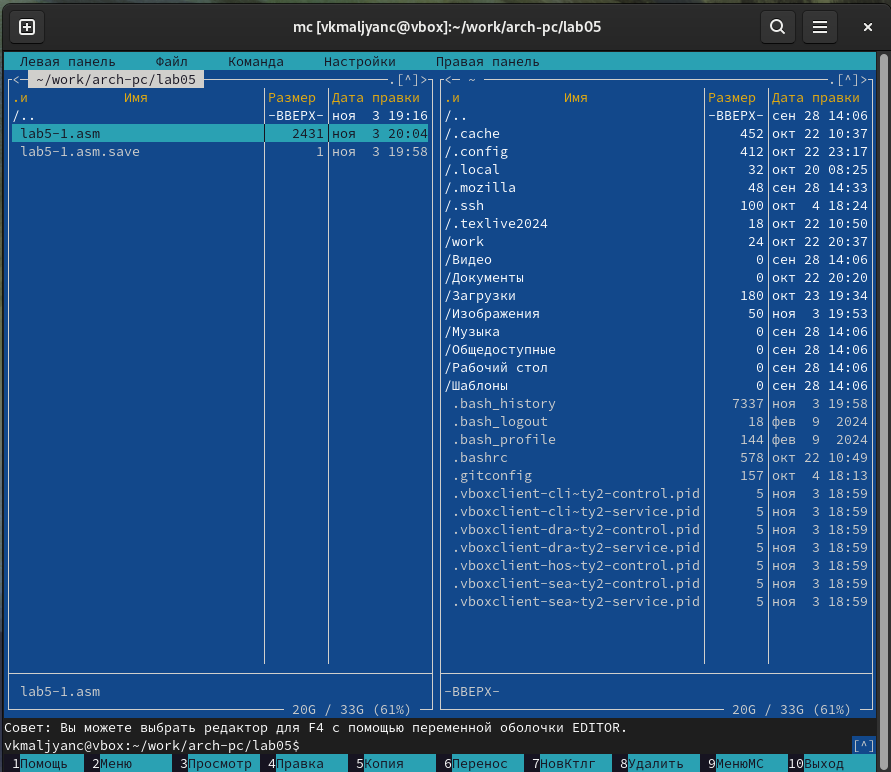


Рис. 9: Выход из редактора

Открываю файл lab5-1.asm для просмотра с помощью функциональной клавиши F3. Убеждаюсь, что файл содержит текст программы (рис. 10).

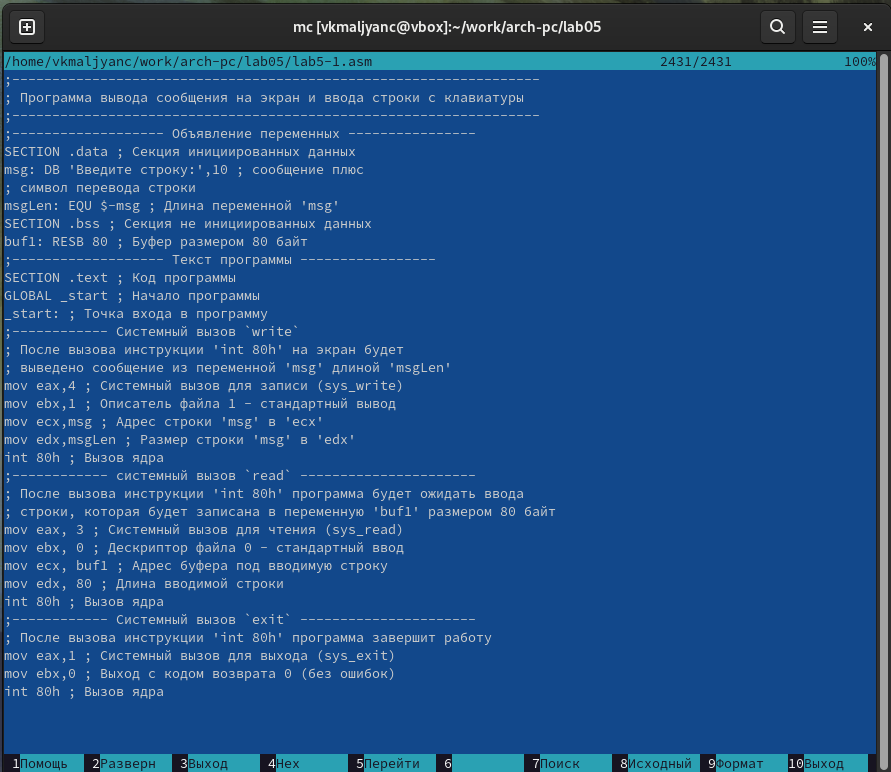


Рис. 10: Просмотр содержимого файла lab5-1.asm

Транслирую текст программы lab5-1.asm в объектный файл (рис. 11).

Рис. 11: Транслирование текста в объектный файл

Рис. 11: Транслирование текста в объектный файл

Передаю объектный файл на обработку компоновщику LD (рис. 12).

Рис. 12: Компоновка объектного файла

Рис. 12: Компоновка объектного файла

Запускаю получившийся исполняемый файл (рис. 13).

Рис. 13: Запуск исполняемого файла

Рис. 13: Запуск исполняемого файла

Скачиваю файл in\_out\_asm со страницы курса в ТУИС, он сохранился в каталог Загрузки (рис. 14).

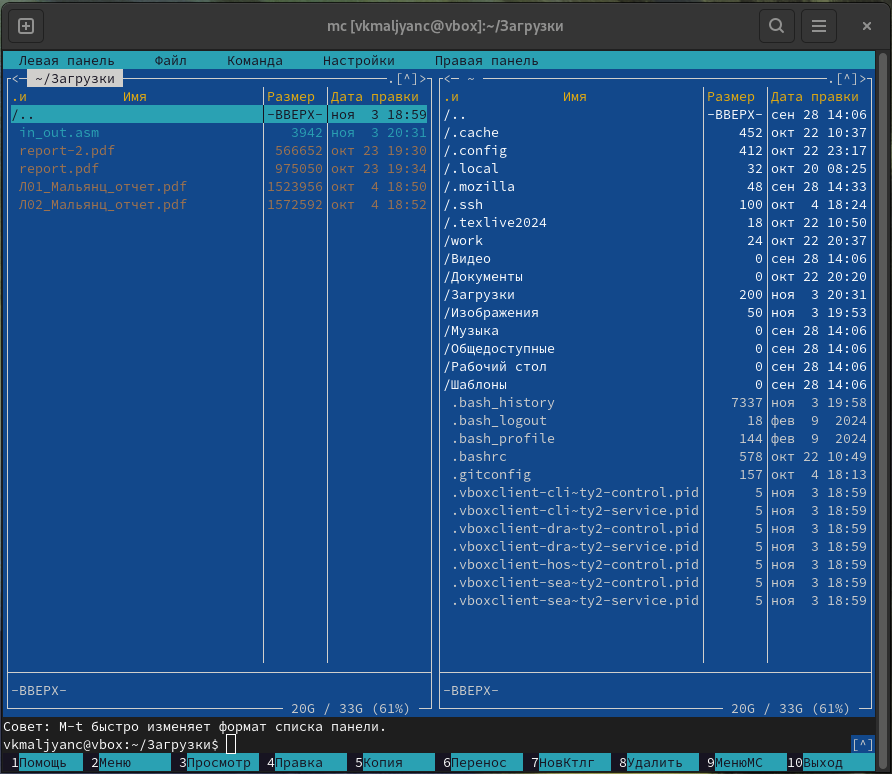


Рис. 14: Скачанный файл in\_out\_asm

Копирую файл in\_out.asm из каталога Загрузки в каталог lab05 с помощью функциональной клавиши F5 (рис. 15).

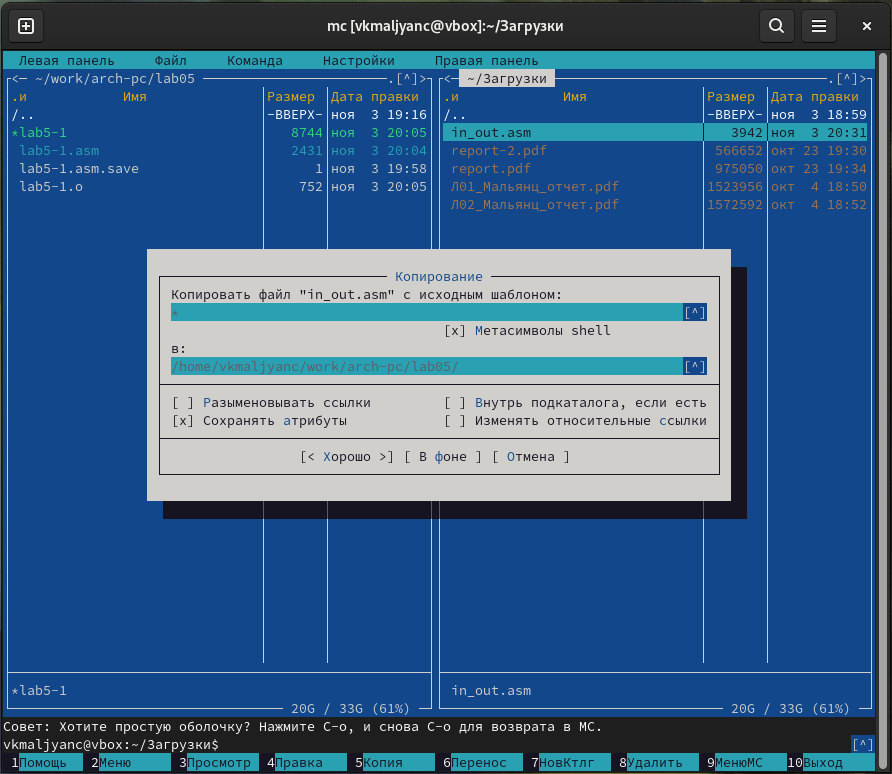


Рис. 15: Копирование файла in\_out.asm в каталог lab05

Создаю копию файла lab5-1.asm с именем lab5-2.asm с помощью функциональной клавиши F6 (рис. 16).

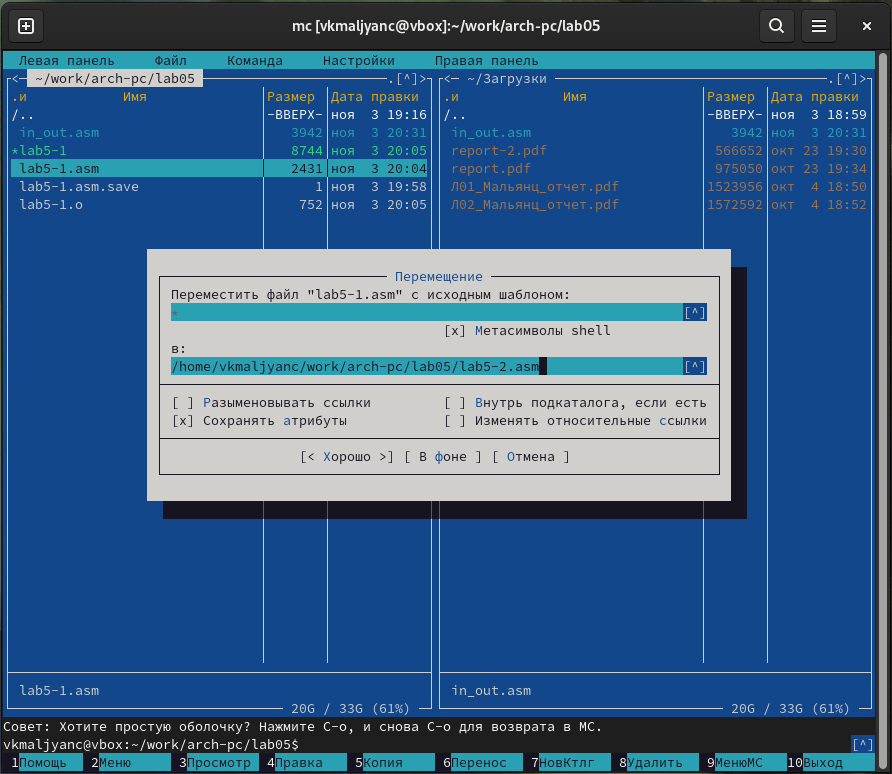


Рис. 16: Создание копии файла lab5-1.asm с именем lab5-2.asm

Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (использую подпрограммы sprintLF, sread и quit) (рис. 17).

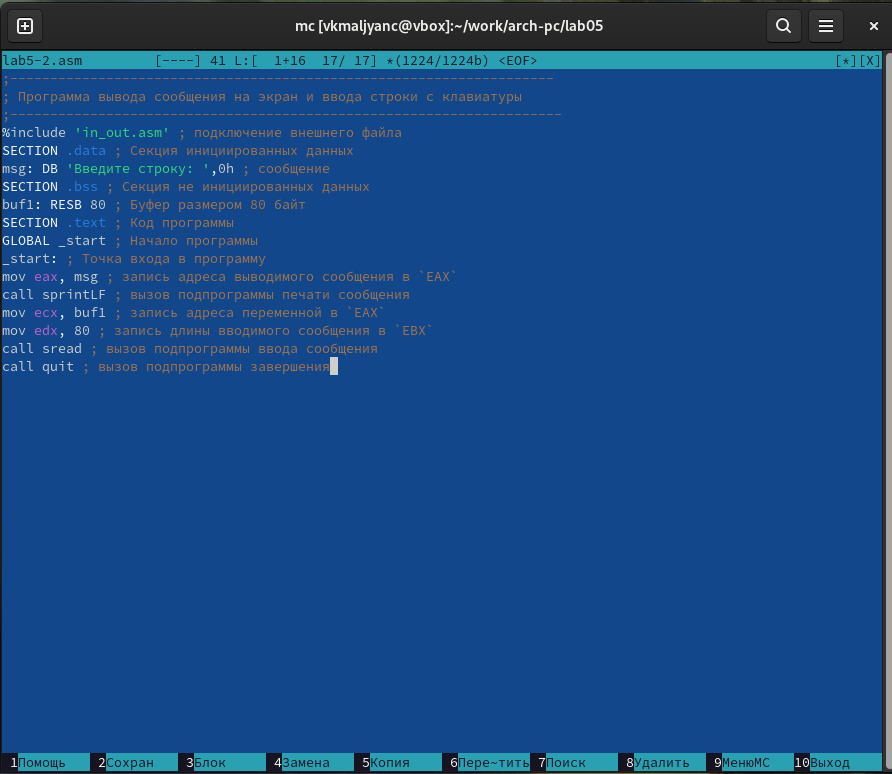


Рис. 17: Редактирование текста программы в файле lab5-2.asm

Транслирую текст программы lab5-2.asm в объектный файл, передаю объектный файл на обработку компоновщику LD, запускаю получившийся исполняемый файл (рис. 18).

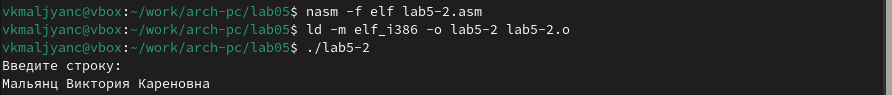


Рис. 18: Исполнение файла

Заменяю подпрограмму sprintLF на sprint в файле lab5-2.asm (рис. 19).

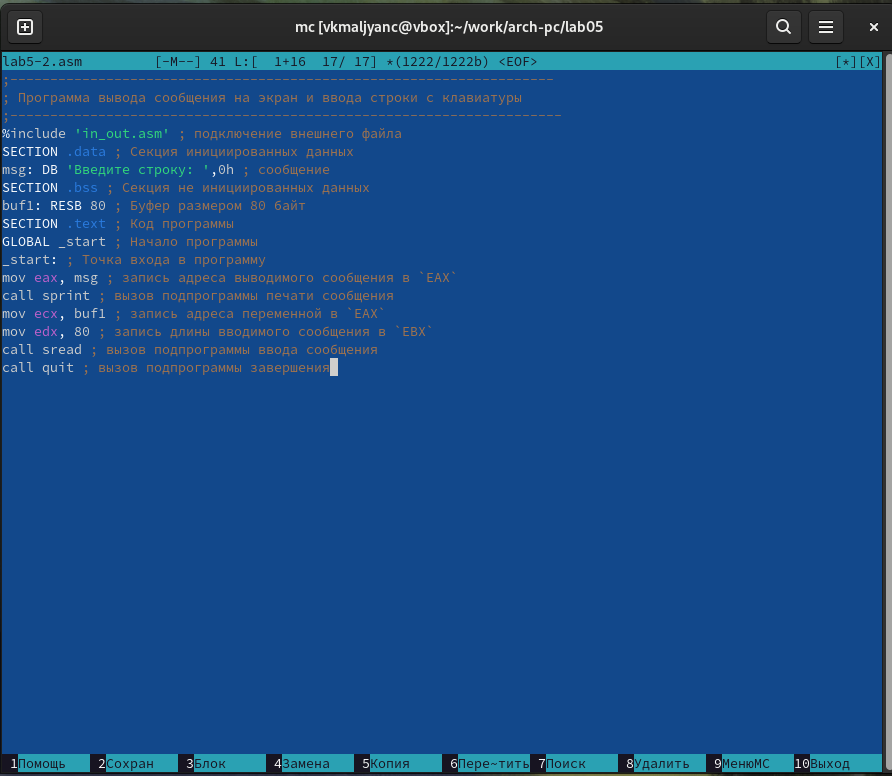


Рис. 19: Редактирование текста программы в файле lab5-2.asm

Транслирую текст программы lab5-2.asm в объектный файл, передаю объектный файл на обработку компоновщику LD, запускаю получившийся исполняемый файл (рис. 20).

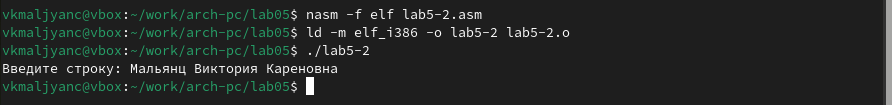


Рис. 20: Исполнение файла

Разница между первым (с подпрограммой sprintLF) и вторым (с подпрограммой sprint) исполняемыми файлами зааключается в том, что в первом исполняемом файле запуск запрашивает ввод с новой строки, а во втором исполняемом файле запуск запрашивает ввод без переноса на новую строку.

## 2.1 Выполнение заданий для самостоятельной работы

1. Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 21).

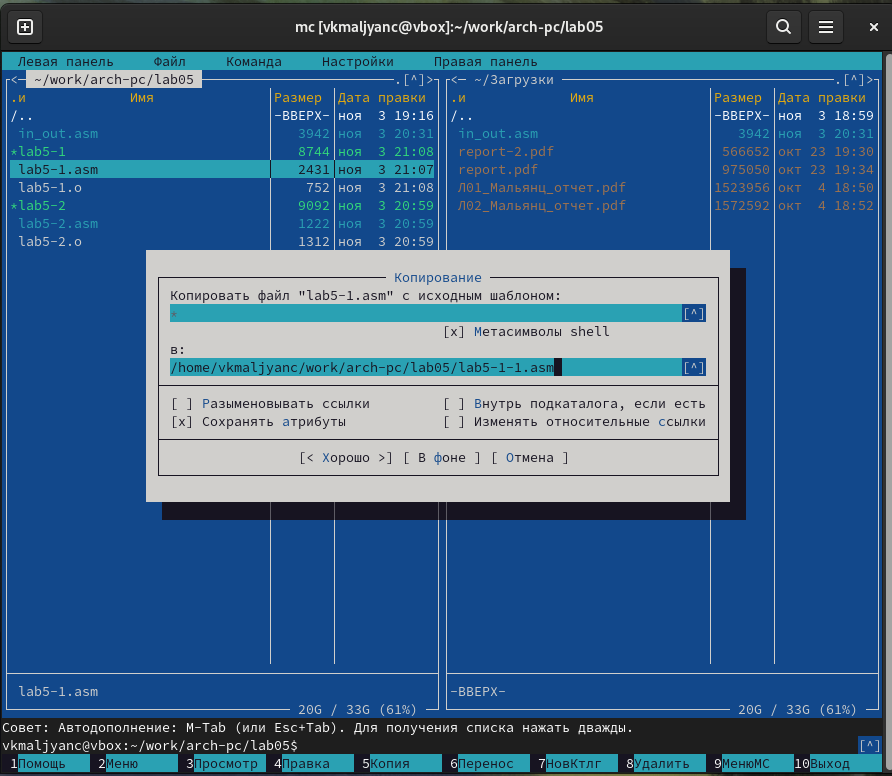


Рис. 21: Создание копии файла lab5-1.asm с именем lab5-1-1.asm

Открываю файл lab5-1-1.asm для редактирования во встроенном редакторе с помощью функциональной клавиши F4. Ввожу текст программы вывода сообщения на экран, ввода строки с клавиатуры и вывода введенной строки на экран (рис. 22). Сохраняю изменения с помощью функциональной клавиши F2. Выхожу из редактора с помощью функциональной клавиши F10.

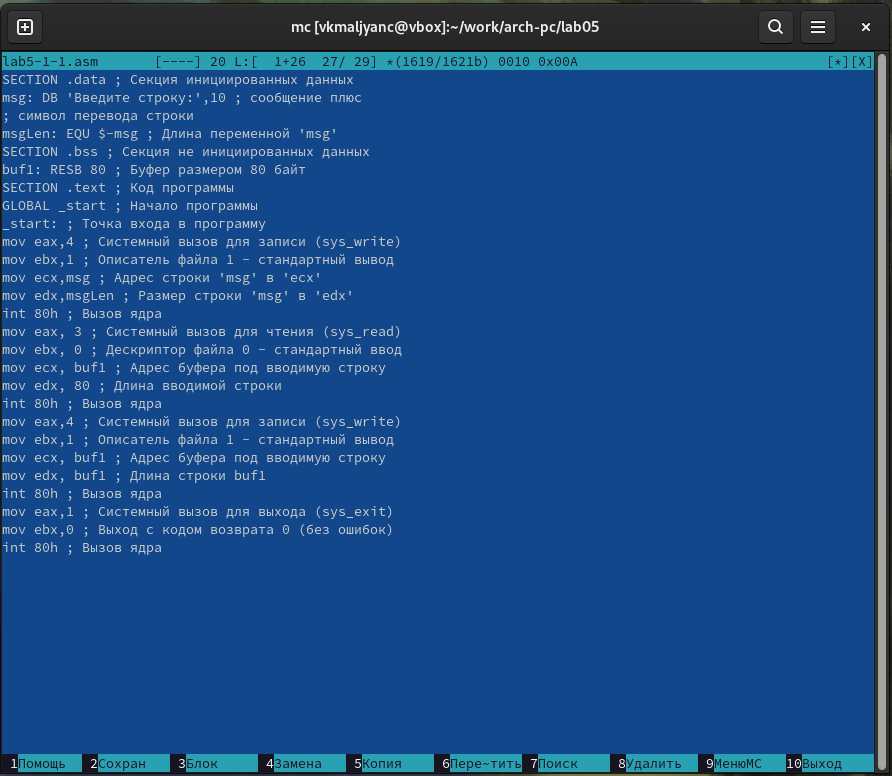


Рис. 22: Ввод текста программы

Листинг программы:

SECTION .data ; Секция инициированных данных msg: DB ‘Введите строку:’,10 ; сообщение плюс ; символ перевода строки 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 буфера под вводимую строку mov edx, buf1 ; Длина 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. Транслирую текст программы lab5-1-1.asm в объектный файл, передаю объектный файл на обработку компоновщику LD, запускаю получившийся исполняемый файл. Ввожу свои ФИО. Программа выводит введенные мною данные (рис. 23).

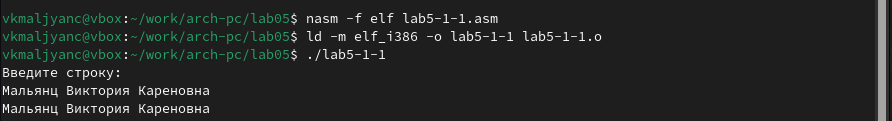


Рис. 23: Исполнение файла

1. Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. 24).

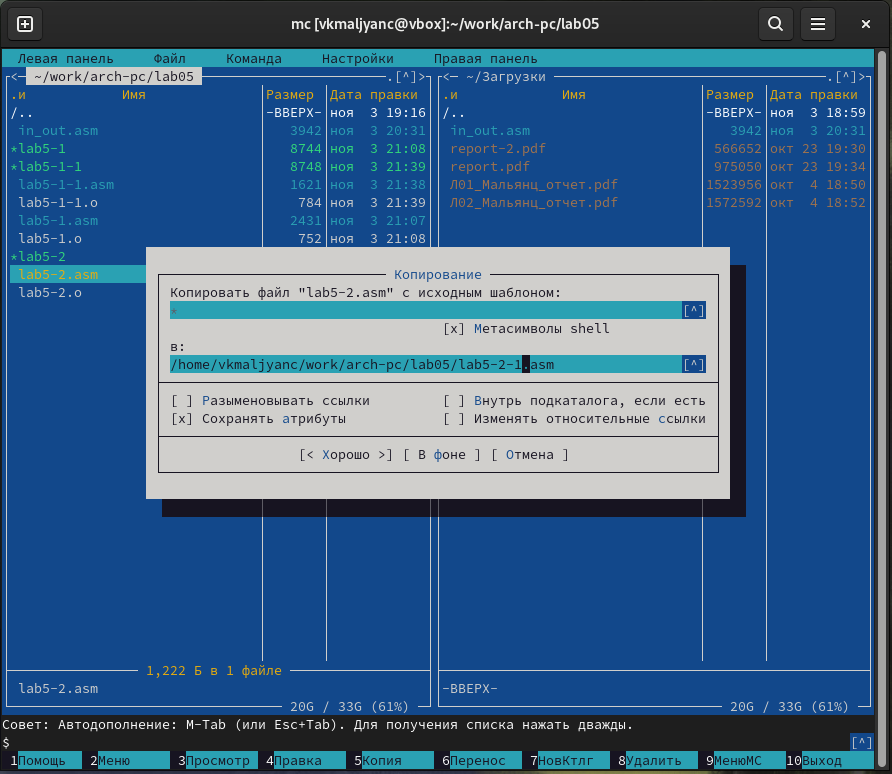


Рис. 24: Создание копии файла lab5-2.asm с именем lab5-2-1.asm

Открываю файл lab5-2-1.asm для редактирования во встроенном редакторе с помощью функциональной клавиши F4. Ввожу текст программы вывода сообщения на экран, ввода строки с клавиатуры и вывода введенной строки на экран (рис. 25). Сохраняю изменения с помощью функциональной клавиши F2. Выхожу из редактора с помощью функциональной клавиши F10.

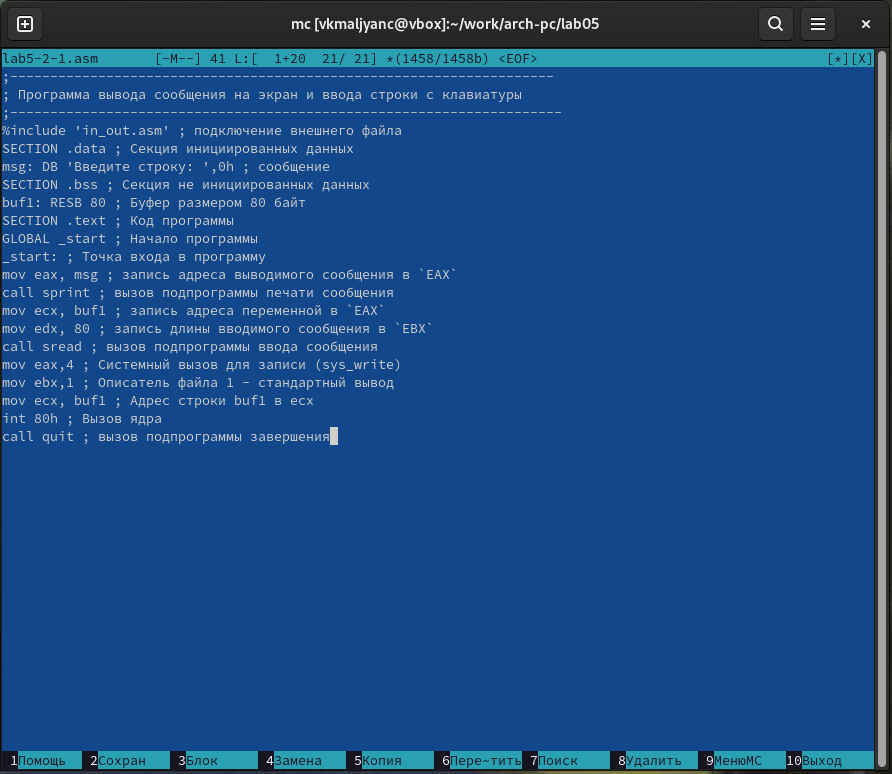


Рис. 25: Ввод текста программы

Листинг программы: ;——————————————————————– ; Программа вывода сообщения на экран и ввода строки с клавиатуры ;——————————————————————— %include ‘in\_out.asm’ ; подключение внешнего файла 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

1. Транслирую текст программы lab5-2-1.asm в объектный файл, передаю объектный файл на обработку компоновщику LD, запускаю получившийся исполняемый файл. Ввожу свои ФИО. Программа выводит введенные мною данные (рис. 26).

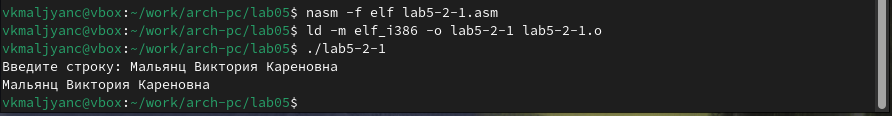


Рис. 26: Исполнение файла

# 3 Выводы

Я приобрела практические навыки работы в Midnight Commander. Освоила инструкции языка ассемблера mov и int.