Лабораторная работа №5

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

Хасанов Тимур

Содержание

# 1 Цель работы

Ознакомиться с программой Midnight commander и освоить написание программ на языке ассемблера с помощью инструкций mov и int

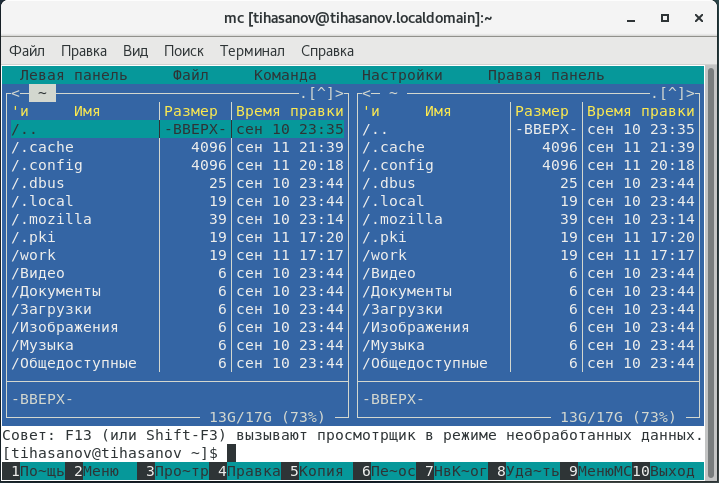
# 2 Выполнение лабораторной работы

Для начала выполнения лабораторной работы нам необходимо открыть Midnight commander с помощью команды mc (Рис. 2.1):

Запуск Midnight commander

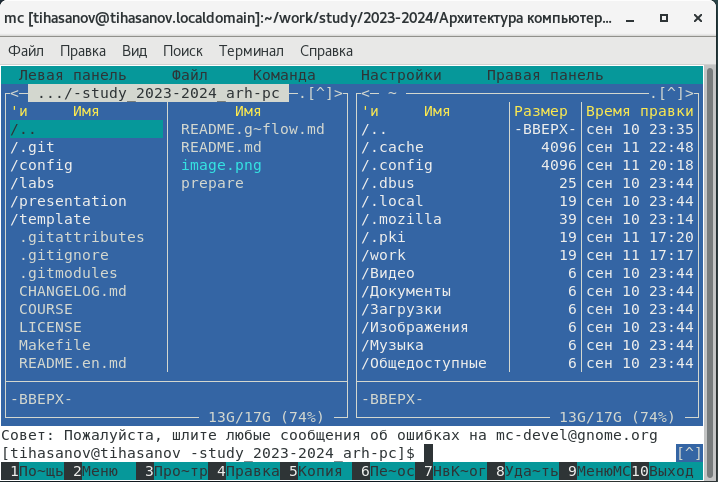
Запуск Midnight commander

После ввода команды мы увидим такой интерфейс (Рис. 2.2):



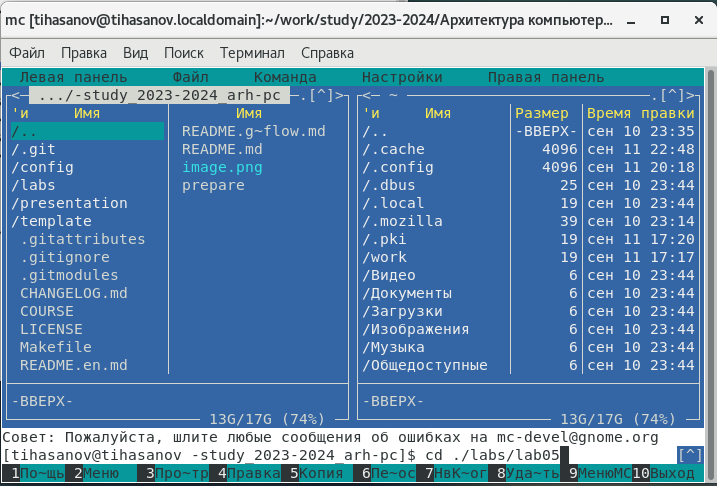
Интерфейс midnight commander

С помощью стрелок и клавиши Enter перейдём в каталог с репозиторием ~/work/study/2023-2024/Архитектура компьютера/-study\_2023-2024\_arh-pc (Рис. 2.3):



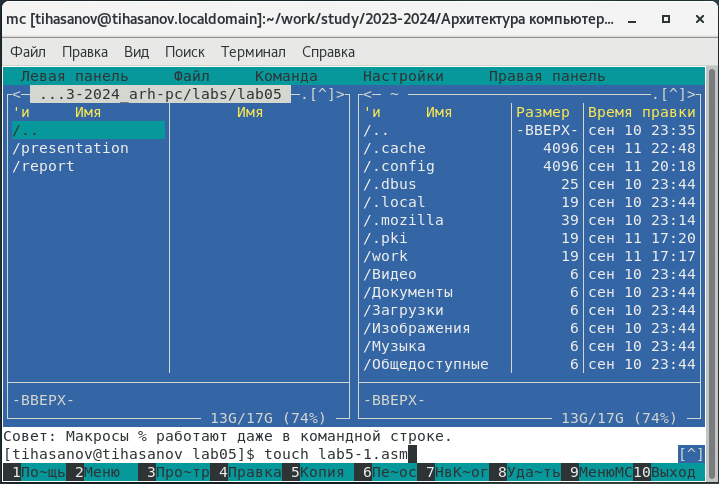
Переход в нужный каталог

Далее с помощью утилиты cd перейдём в каталог в котором будем работать (Рис. 2.4):



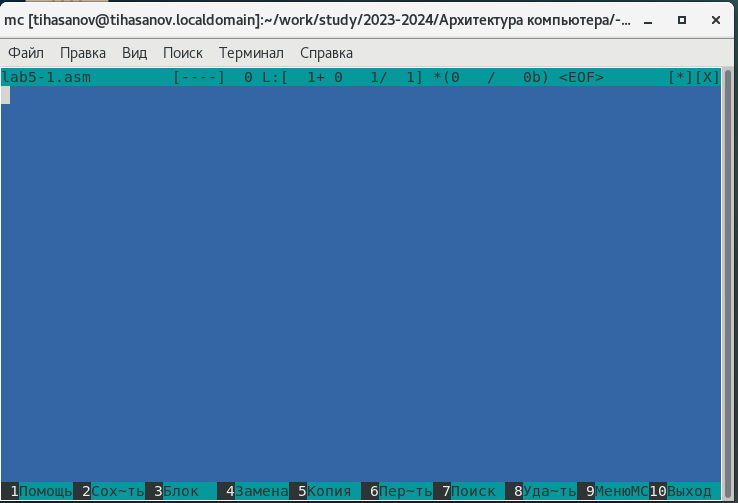
Создание папки

Теперь с помощью команды touch создадим файл lab5-1.asm (Рис. 2.5):



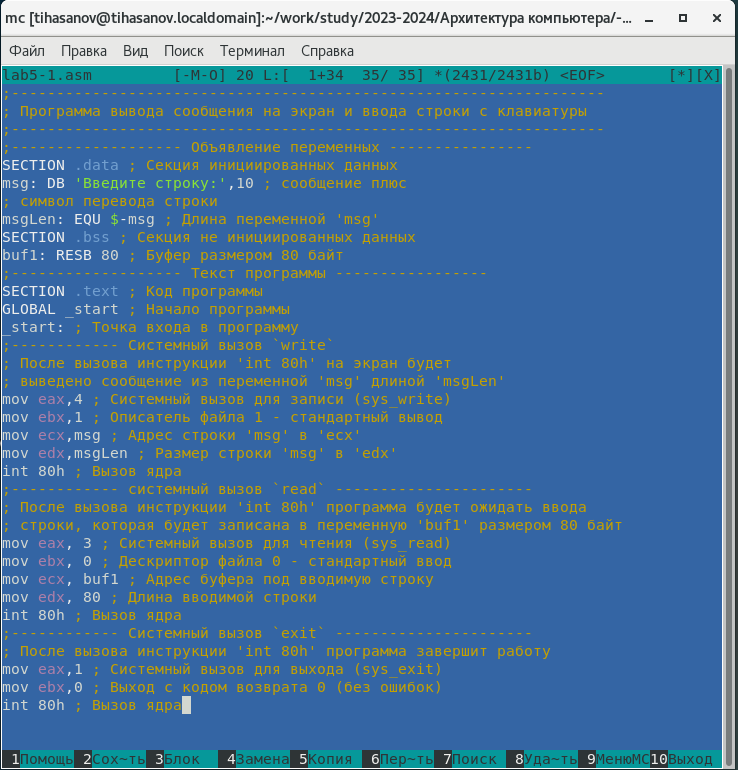
Создание файла lab5-1.asm с помощью команды touch прямо в mc

Теперь с помощью клавиши F4 откроем только что созданный файл. Файл по умолчанию будет открыт в редакторе mcedit (Рис. 2.6):



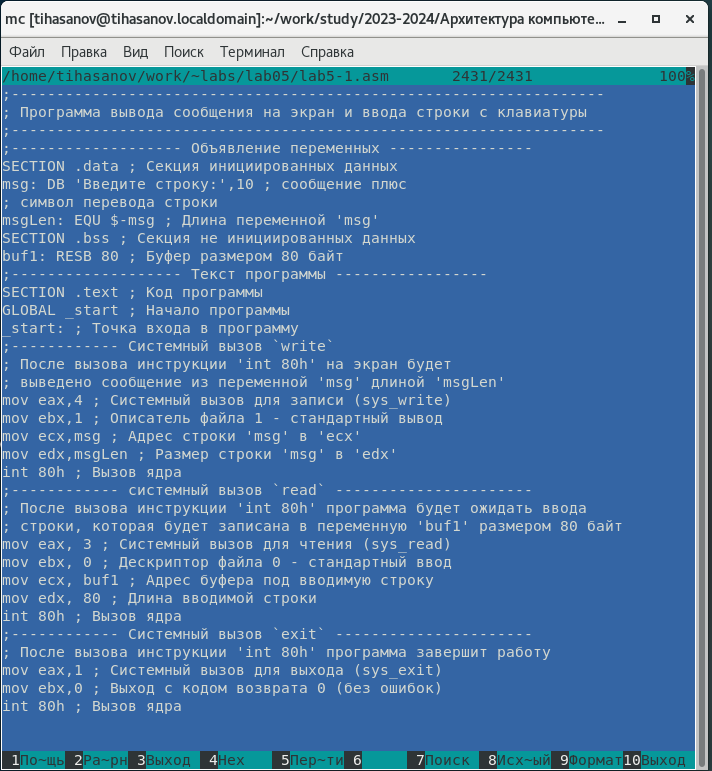
Выбор текстового редактора

Теперь отредактируем файл и поместим в него следующий код (Рис. 2.7):



Редактирование файла lab5-1.asm

Теперь сохраним его (клавишей F2, согласившись с сохранением, и F10 чтобы выйти из редактора) и с помощью F3 откроем для просмотра, чтобы убедиться, что он сохранился корректно (Рис. 2.8):



Проверка успешного редактирования

Теперь скомпилируем его (Рис. 2.9):

Компиляция файла с помощью nasm

Компиляция файла с помощью nasm

И соберём (Рис. 2.10):

Сборка исполняемого файла с помощью ld

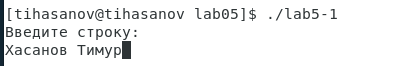
Сборка исполняемого файла с помощью ld

После этого запустим получившийся исполняемый файл (Рис. 2.11):

Запуск исполняемого файла

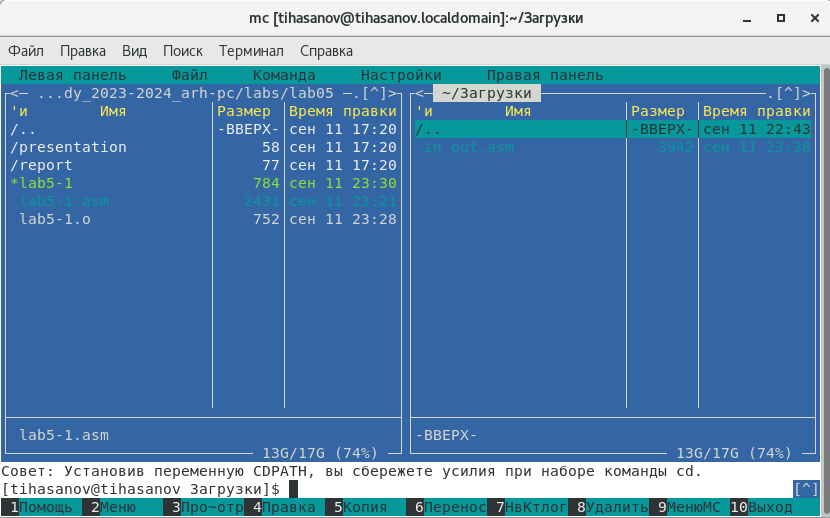
Запуск исполняемого файла

Теперь введём ФИ (Рис. 2.12):



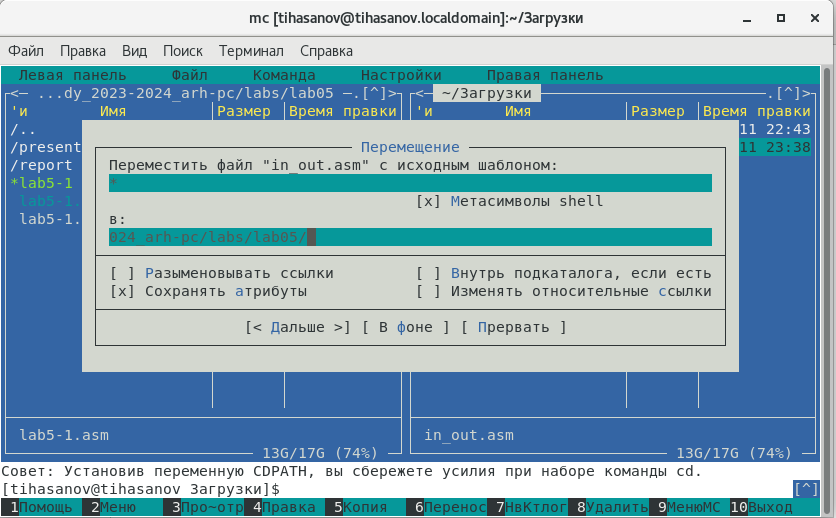
Взаимодействие с программой

После нажатия Enter программа завершится и ничего не произойдёт. Теперь скачаем файл in\_out.asm и откроем папку с ним в правой панели (Рис. 2.13):



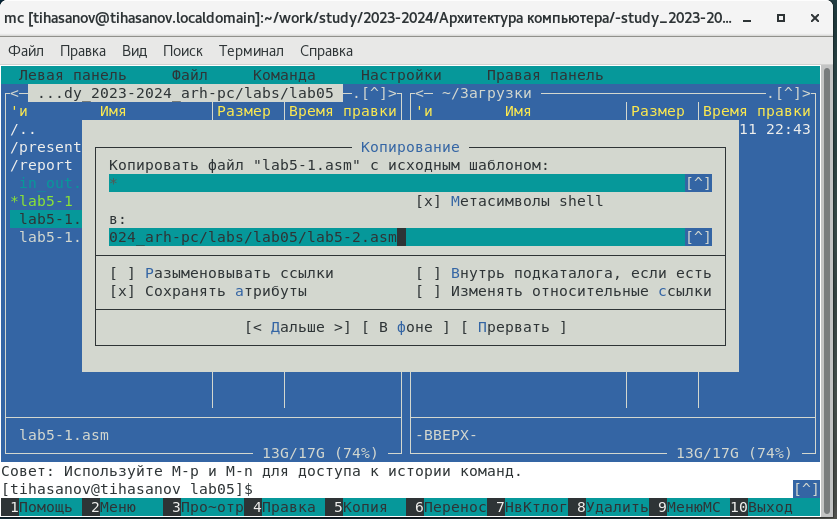
Открытие папки с файлом in\_out.asm в правой панели

Переместим его в нашу рабочую папку с помощью F6 (Рис. 2.14):



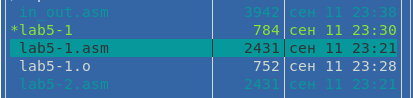
Перемещение файла c помощью F6

Теперь сделаем копию файла lab5-1.asm с помощью команды F5. Назовём копию lab5-2.asm (Рис. 2.15):



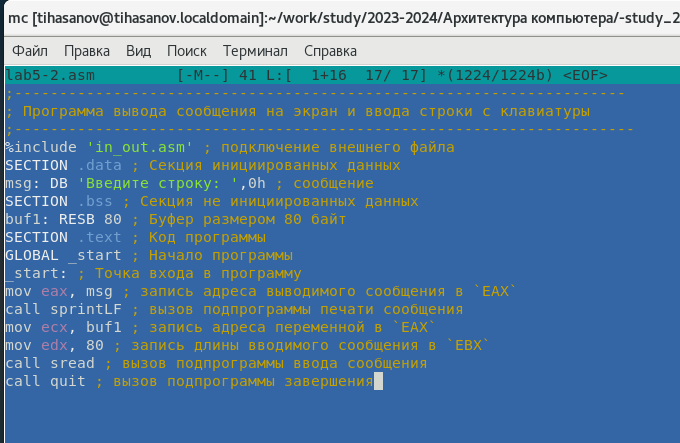
Копирование файла c помощью F5

Теперь наша папка выглядит следующим образом (Рис. 2.16):



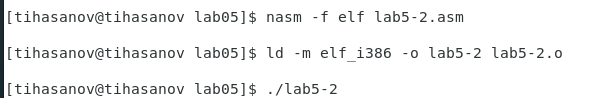
Текущий вид рабочей папки

Откроем в текстовом редакторе файл lab5-2.asm и напишем туда следующий код (Рис. 2.17):



Редактирование файла lab5-2.asm

После чего создадим исполняемый файл с помощью nasm и ld (Рис. 2.18):



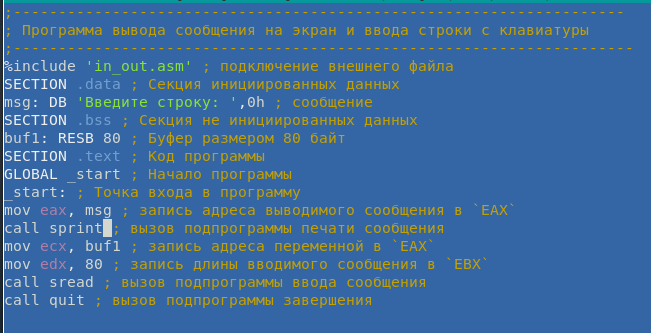
Создание исполняемого файла

Запустим созданный файл (Рис. 2.19):

Запуск исполняемого файла

Запуск исполняемого файла

Он работает также, как и файл lab5-1, но использует для работы сторонний файл. Попробуем теперь вместо команды sprintLF использовать просто команду sprint (Рис. 2.20):



Изменение файла lab5-2.asm

Точно также соберём исполняемый файл и запустим его (Рис. 2.21):

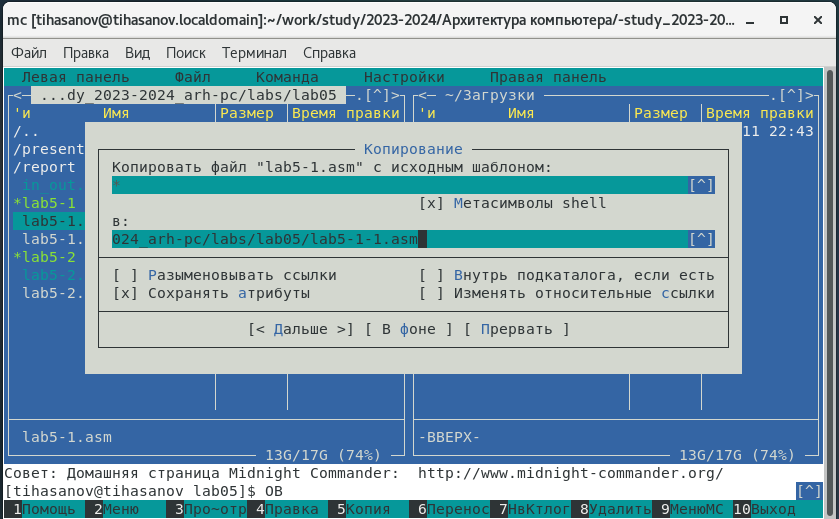
Запуск изменённого файла

Запуск изменённого файла

Как мы видим, теперь нет переноса на следующую строку. Этим и отличаются команды sprintLF от sprint. Первая добавляет перенос после текста, а вторая нет

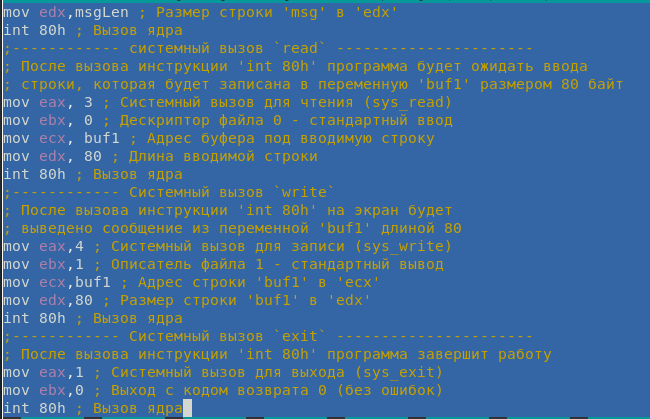
# 3 Выполнение задания для самостоятельной работы

Теперь создадим с помощью F5 копию файла lab5-1.asm (Рис. 3.1):



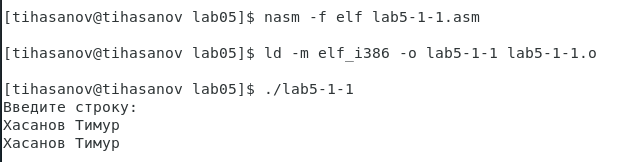
Создание копии файла lab5-1.asm

Изменим копию так, чтобы она выводила тот текст, который получила на ввод. Для этого перед системным вызовом exit вставим текст с системным вызовом write. Он очень похож на системный вызов write, который уже был в коде, но есть несколько отличий. Так, мы перемещаем адрес строки buf1 в ecx и размер строки buf1 (80) в edx (Рис. 3.2):



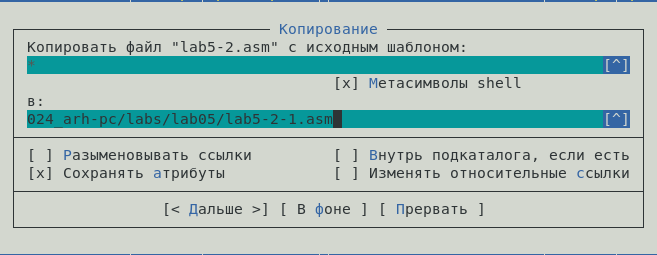
Изменение файла lab5-1-1.asm

Сохраним изменения, создадим исполняемый файл, запустим программу и проверим, что всё работает (Рис. 3.3):



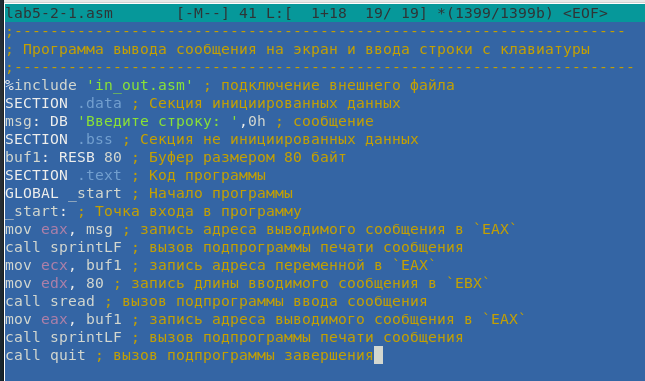
Проверка работы программы

Теперь создадим с помощью F5 копию файла lab5-2.asm (Рис. 3.4):



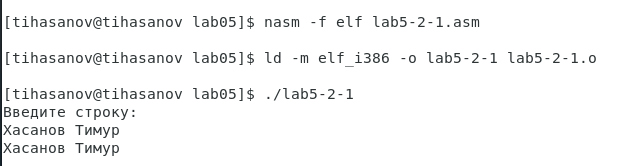
Создание копии файла lab5-2.asm

Теперь сделаем так, чтобы этот код также выводил тот текст, что получит на ввод. Для этого перед последней строкой добавим строчку, которая записывает в eax адрес buf1, а также строчку, которая вызывает подпрограмму sprintLF (Рис. 3.5):



Изменение файла lab5-2-1.asm

Теперь создадим исполняемый файл, запустим программу и убедимся, что она работает (Рис. 3.6):



Создание исполняемого файла

# 4 Выводы

В результате выполнения работы были получены навыки работы с Midnight commander, а также навыки написания простых программ ввода-вывода на языке ассемблера