Лабораторная работа №5

Дисциплина: Архитектура компьютера

Малюга Валерия Васильевна

Содержание

[1 Цель работы 1](#_Toc150464876)

[2 Задание 1](#_Toc150464877)

[3 Теоретическое введение 2](#_Toc150464878)

[3.1 Основы работы с Midnight Commander 2](#_Toc150464879)

[3.2 Структура программы на языке ассемблера NASM 2](#_Toc150464880)

[3.3 Описание инструкции mov 3](#_Toc150464881)

[3.4 Описание инструкций int 3](#_Toc150464882)

[4 Выполнение лабораторной работы 3](#_Toc150464883)

[4.1 Основы работы с Midnight Commander 3](#_Toc150464884)

[4.2 Подключение внешнего файла in\_out.asm 6](#_Toc150464885)

[4.3 Задание для самостоятельной работы 9](#_Toc150464886)

[5 Выводы 12](#_Toc150464887)

# 1 Цель работы

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

# 2 Задание

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

# 3 Теоретическое введение

## 3.1 Основы работы с Midnight Commander

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной.  
В Midnight Commander используются функциональные клавиши *F1 — F10*, к которым привязаны часто выполняемые операции (табл. [1](#tbl:std-dir))

Table 1: Функциональные клавиши Midnight Commander

|  |  |
| --- | --- |
| Функциональные клавиши | Выполняемое действие |
| F1 | вызов контекстно-зависимой подсказки |
| F2 | вызов меню, созданного пользователем |
| F3 | просмотр файла, на который указывает подсветка в активной панели |
| F4 | вызов встроенного редактора для файла, на который указывает подсветка в активной панели |
| F5 | копирование файла (группы файлов из каталога), отображаемого в активной панели, в каталог, отображаемый на второй панели |
| F6 | перенос файла (группы файлов из каталога), отображаемого в активной панели, в каталог, отображаемый на второй панели |
| F7 | создание подкаталога в каталоге, отображаемом в активной панели |
| F8 | удаление файла (подкаталога) или группы отмеченных файлов |
| F9 | вызов основного меню программы |
| F10 | выход из программы |

## 3.2 Структура программы на языке ассемблера NASM

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти:

* DB (define byte) — определяет переменную размером в 1 байт;
* DW (define word) — определяет переменную размеров в 2 байта (слово);
* DD (define double word) — определяет переменную размером в 4 байта (двойное слово);
* DQ (define quad word) — определяет переменную размером в 8 байт (учетверённое слово);
* DT (define ten bytes) — определяет переменную размером в 10 байт.
* Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти.

## 3.3 Описание инструкции mov

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

mov dst,src

Здесь операнд dst — приёмник, а src — источник.  
В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const).

## 3.4 Описание инструкций int

Инструкция языка ассемблера int предназначена для вызова прерывания с указанным номером.

int n

Здесь n — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра sys\_calls n=80h (принято задавать в шестнадцатеричной системе счисления).

# 4 Выполнение лабораторной работы

## 4.1 Основы работы с Midnight Commander

Открыла Midnight Commander с помощью команды mc. Пользуясь клавишами ↑ , ↓ и Enter перешла в каталог ~/work/arch-pc, созданный при выполнении лабораторной работы №4. С помощью функциональной клавиши F7 создала папку lab05 (рис. [1](#fig:001)). Перешла в созданный каталог и, пользуясь строкой ввода и командой touch, создала файл lab5-1.asm (рис. [2](#fig:003))

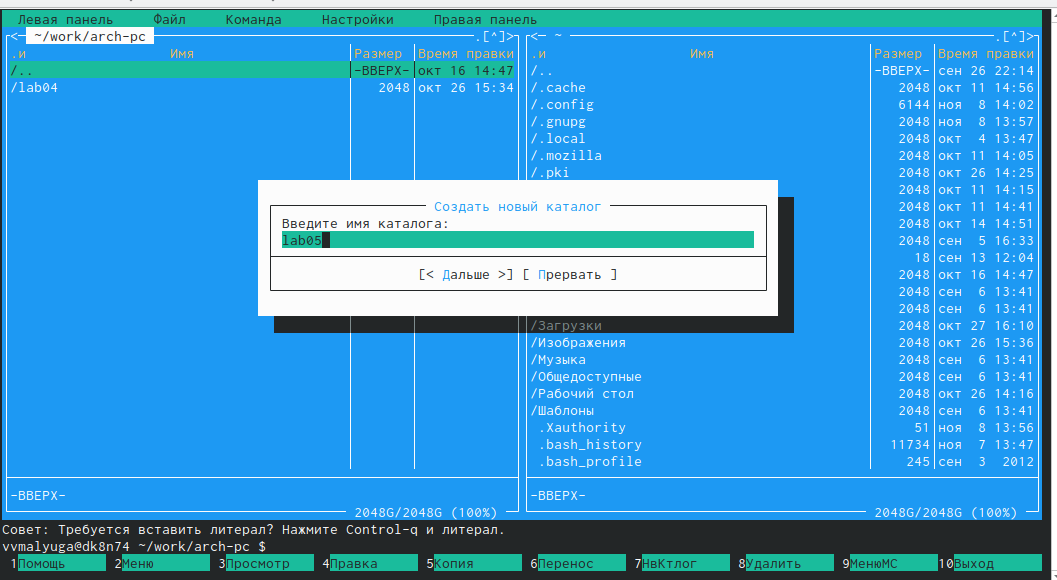


Рис. 1: Создание папки lab05

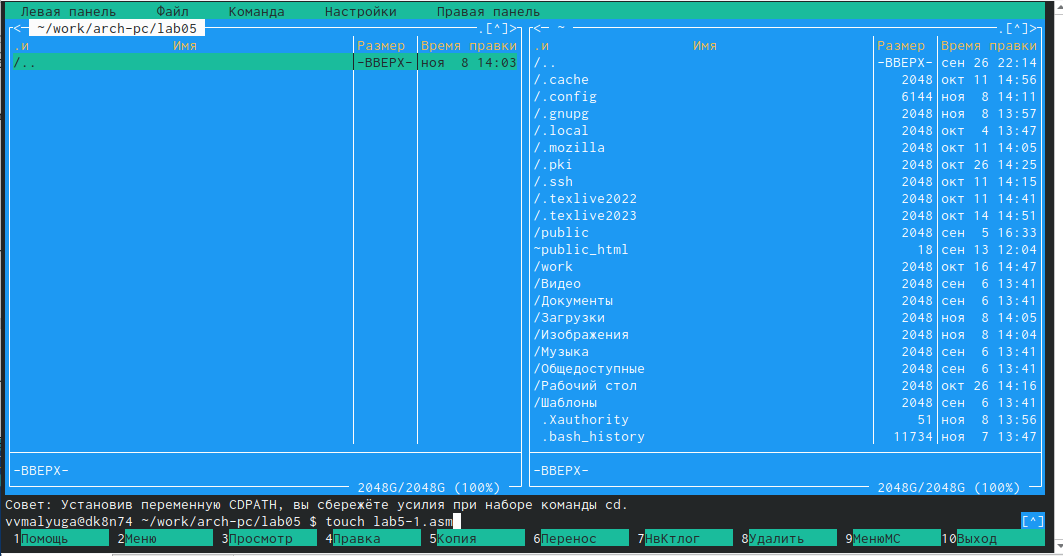


Рис. 2: Создание файла lab5-1.asm

С помощью функциональной клавиши F4 откройте файл lab5-1.asm для редактирования во встроенном редакторе. Введите текст программы из листинга 5.1, сохранила изменения и закрыла файл (рис. [3](#fig:005)). С помощью функциональной клавиши F3 открыла файл lab5-1.asm для просмотра. Убедилась, что файл содержит текст программы (рис. [4](#fig:002)).

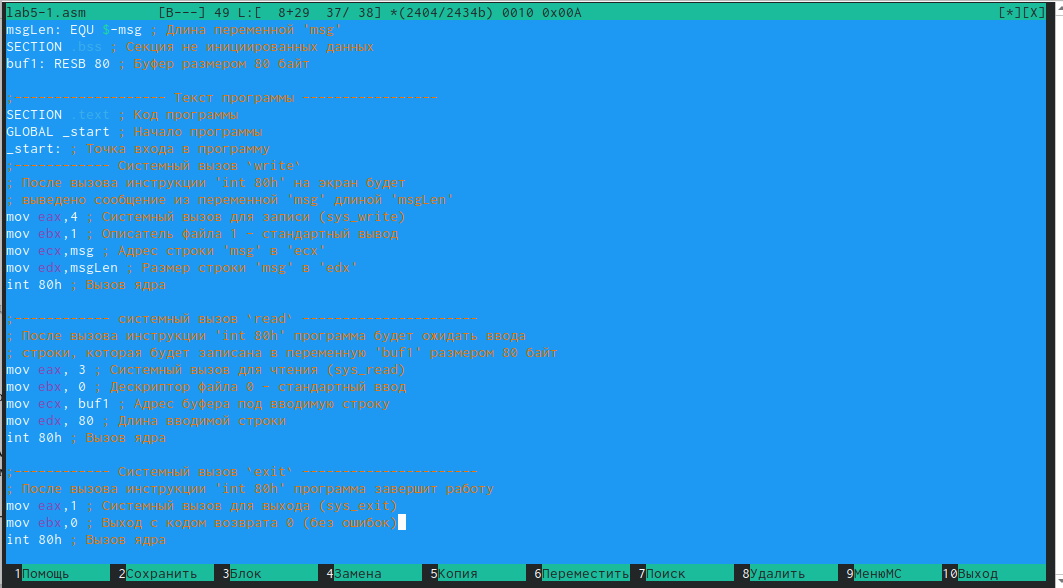


Рис. 3: Ввод текста программы из листинга

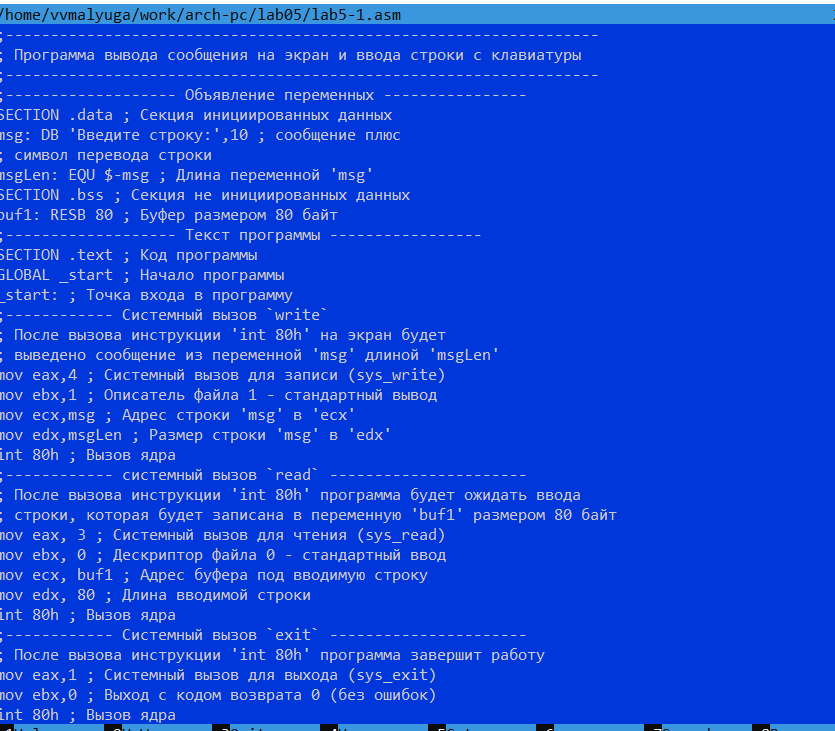


Рис. 4: Проверка содержимого программы

Оттранслировала текст программы lab5-1.asm в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл. Программа выводит строку ‘Введите строку:’ и ожидает ввода с клавиатуры. На запрос ввела свои ФИО (рис. [5](#fig:006)).

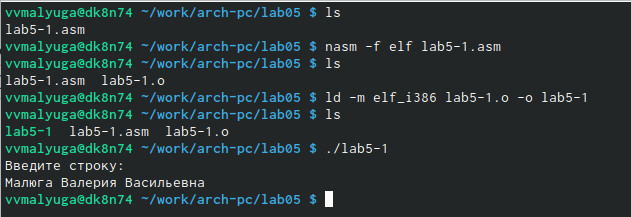


Рис. 5: Трансляция, компоновка объектного файла и запуск программы

## 4.2 Подключение внешнего файла in\_out.asm

Скачала файл in\_out.asm со страницы курса в ТУИСе. С помощью функциональной клавиши F5 скопировала файл in\_out.asm из каталога Загрузки в созданный каталог lab05 (рис. [6](#fig:007)).

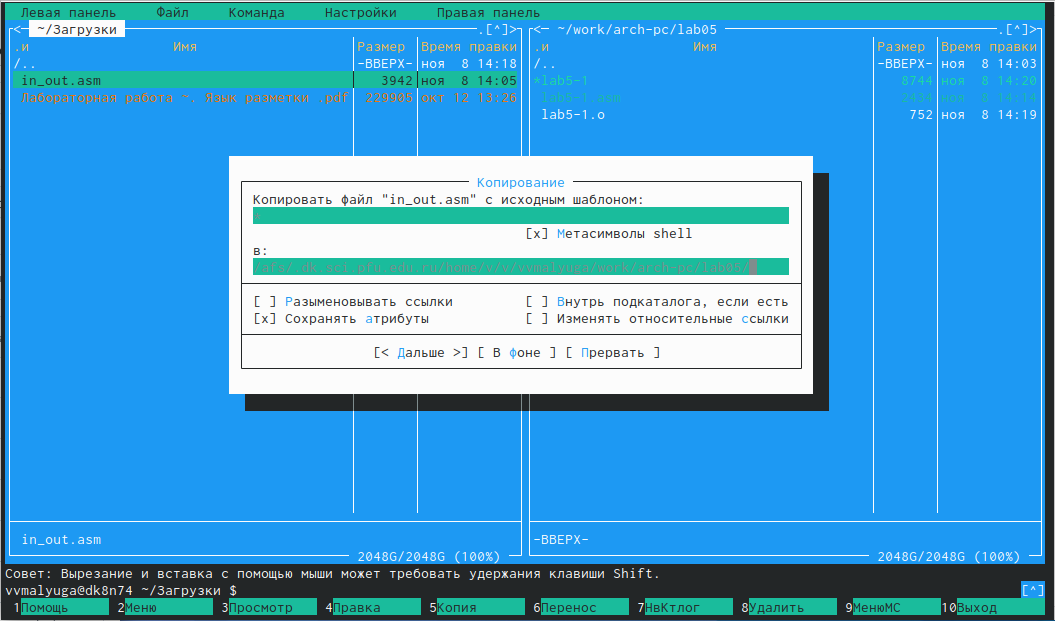


Рис. 6: Копирование файла

С помощью функциональной клавиши F5 скопировала файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне прописала имя для копии файла (рис. [7](#fig:080)).

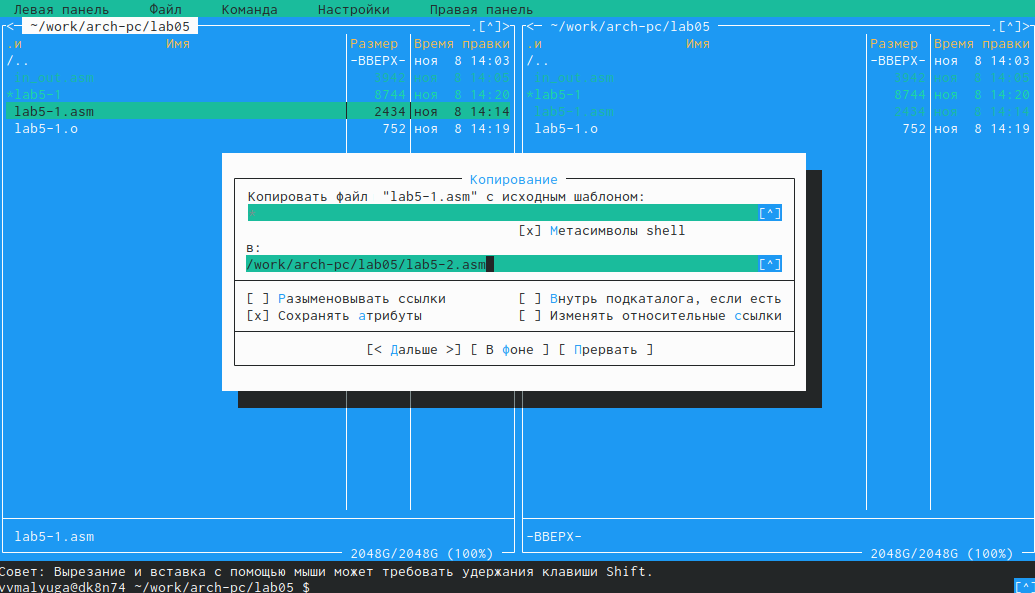


Рис. 7: Копирование файла

Изменила содержимое файла lab5-2.asm во встроенном редакторе (рис. [8](#fig:009)), чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm.

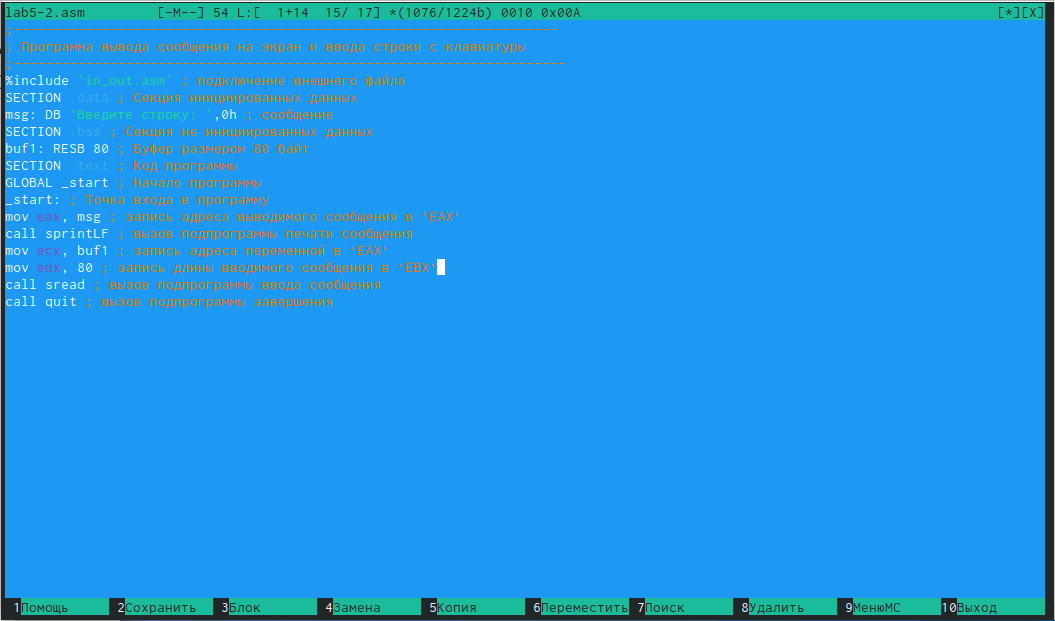


Рис. 8: Редактирование файла

Оттранслировала текст программы файла в объектный файл командой nasm -f elf lab5-2.asm. Создался объектный файл lab5-2.o. Выполнила компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab5-2 lab5-2.o Создался исполняемый файл lab5-2. Запустила исполняемый файл (рис. [[9](#fig:010)]).

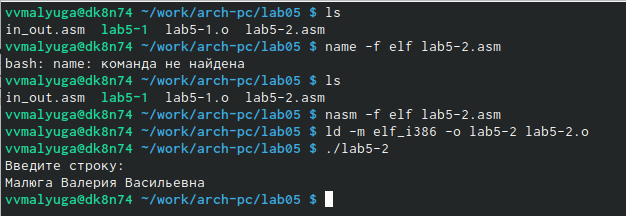


Рис. 9: Исполнение файла

Открыла файл lab5-2.asm для редактирования функциональной клавишей F4. Изменила в нем подпрограмму sprintLF на sprint. Сохранила изменения и открыла файл для просмотра, чтобы проверить сохранение действий (рис. [10](#fig:011)).

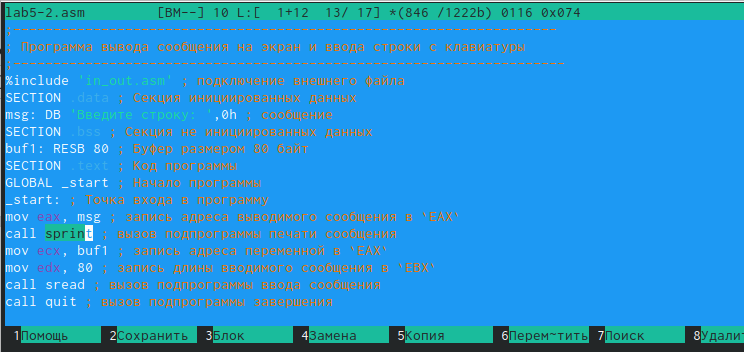


Рис. 10: Отредактированный файл

Разница между первым исполняемым файлом lab5-1 и вторым lab5-2 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 различие между подпрограммами sprintLF и sprint (рис. [11](#fig:012)).

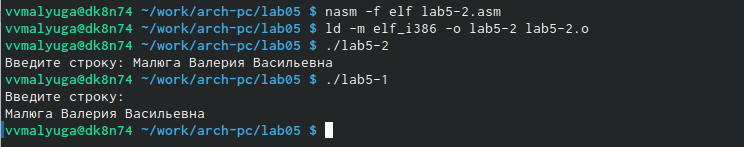


Рис. 11: Вывод программ

## 4.3 Задание для самостоятельной работы

1. Создала копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. [[12](#fig:013)]).

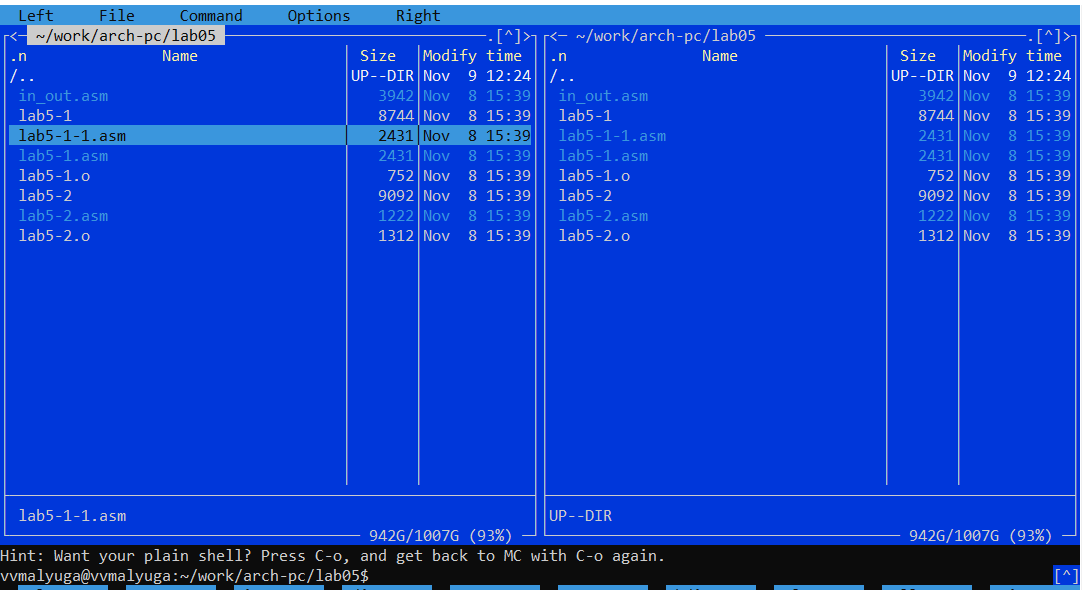


Рис. 12: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменила программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. [[13](#fig:014)]).

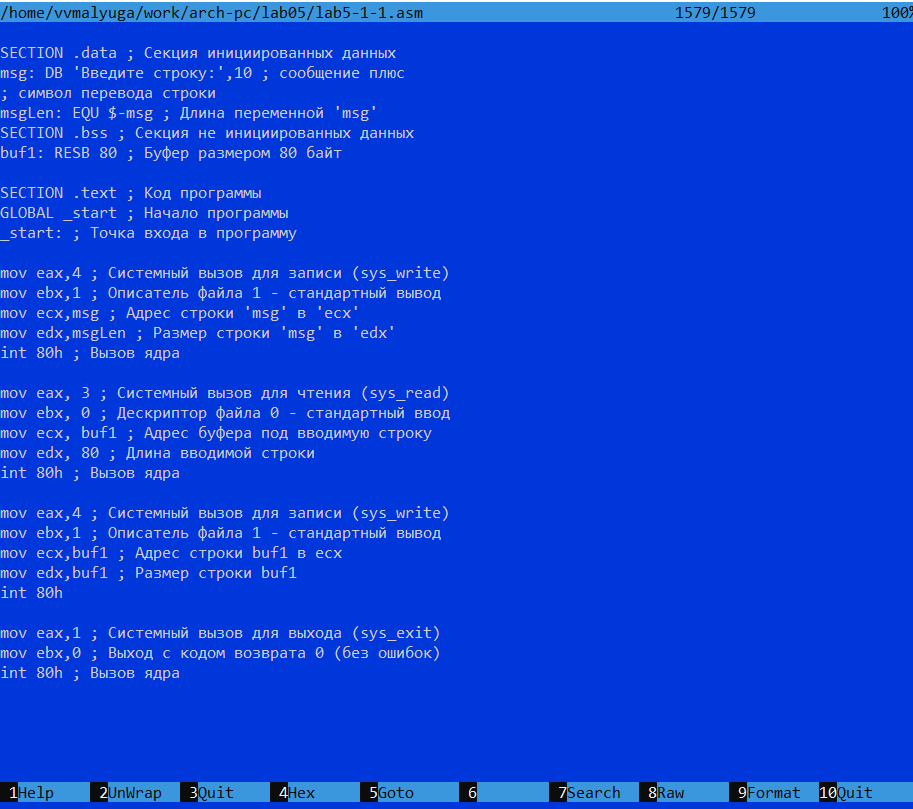


Рис. 13: Редактирование файла

1. Создала объектный файл lab5-1-1.o, отдала его на обработку компоновщику, получила исполняемый файл lab5-1-1. Запустила полученный исполняемый файл. Программа запросила ввод, ввела свою фамилию, далее программа вывела введеную мной фамилию (рис. [[14](#fig:015)]).

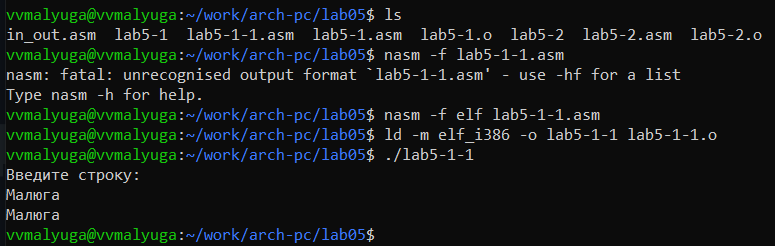


Рис. 14: Исполнение файла

1. Создала копию файла lab5-2.asm с именем lab5-2-2.asm с помощью функциональной клавиши F5 (рис. [[15](#fig:017)]).

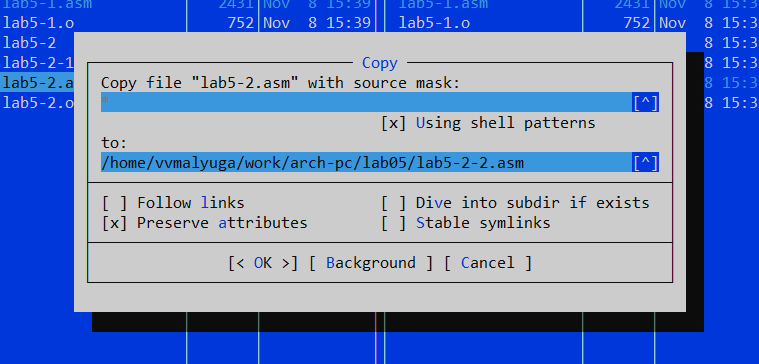


Рис. 15: Копирование файла

С помощью функциональной клавиши F4 открывала созданный файл для редактирования. Изменила программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. [[16](#fig:018)]).

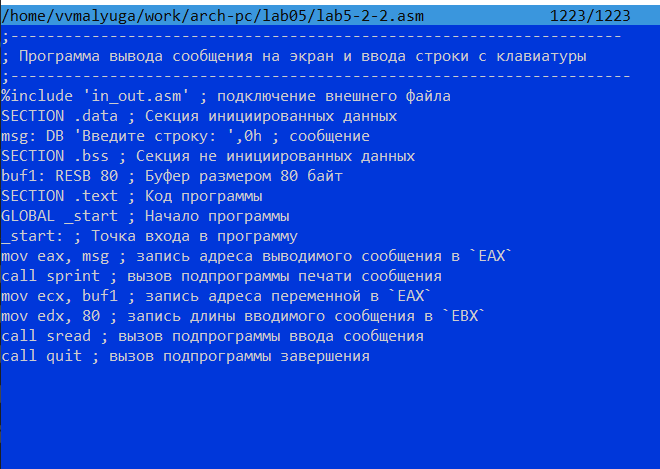


Рис. 16: Редактирование файла

1. Создала объектный файл lab5-2-2.o, отдала его на обработку компоновщику, получила исполняемый файл lab5-2-2. Запустила полученный исполняемый файл. Программа запросила ввод без переноса на новую строку, ввожу свою фамилию, далее программа вывела введеную мной фамилию (рис. [[17](#fig:019)]).

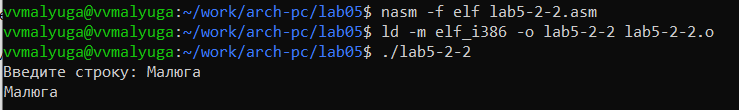


Рис. 17: Исполнение файла

# 5 Выводы

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