Отчёт по лабораторной работе №5

Дисциплина: Архитектура компьютера

Кижваткина Анна Юрьевна

Содержание

# 1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

# 2 Задание

1. Порядок выполнения лабораторной работы.
2. Выполнение самостоятельной работы.

# 3 Теоретическое введение

Midnight Commander, также известный как mc, это программа, которая предоставляет возможность просматривать структуру каталогов и выполнять основные операции управления файловой системой, то есть, mc является файловым менеджером. Midnight Commander облегчает работу с файлами и делает ее более удобной. Программа написана на языке ассемблера NASM и обычно состоит из трех разделов: код программы (SECTION .text), инициализированные данные (SECTION .data) и неинициализированные данные (SECTION .bss). Для объявления данных в разделе .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают значения, которые должны храниться в этой памяти. Эти директивы используются для объявления переменных и массивов, а для строк обычно применяется директива DB из-за специфики хранения данных в оперативной памяти. Инструкция mov в языке ассемблера используется для копирования данных из одного места в другое.

# 4 Выполнение лабораторной работы

1. Порядок выполнения лабораторной работы.

Открываем mc. (рис. 1)

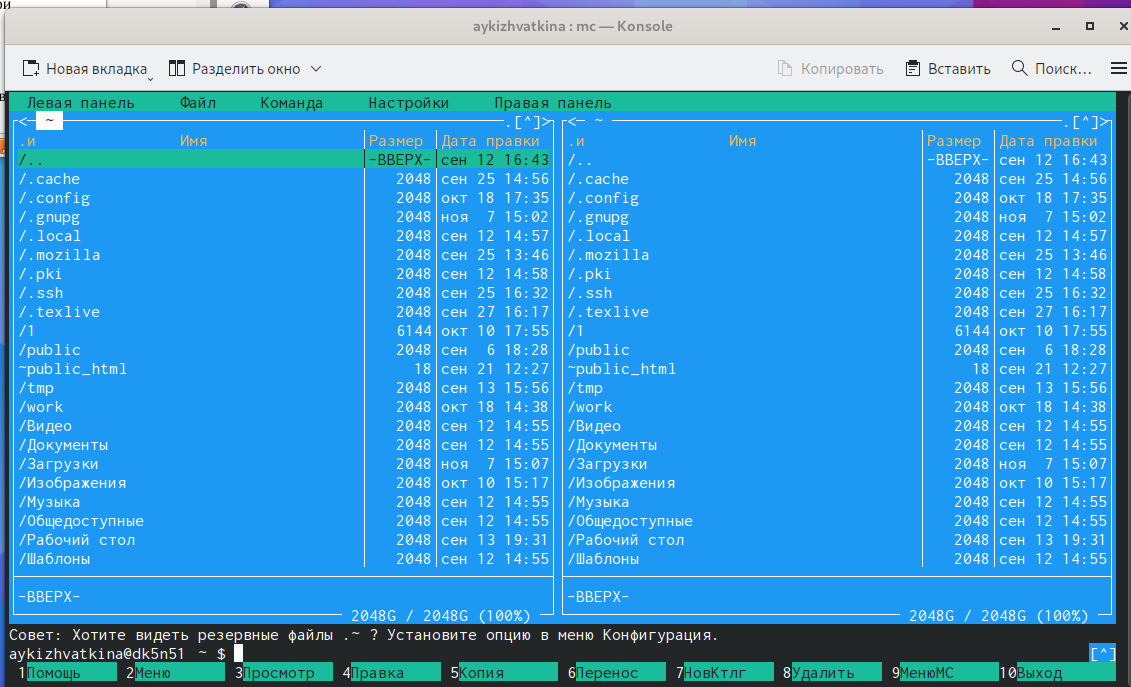


Рис. 1: Окно mc.

Используя клавиши вверх, вниз и Enter переходим в каталог ~/work/arch-pc, созданный при выполнении работы №4. (рис. 2)

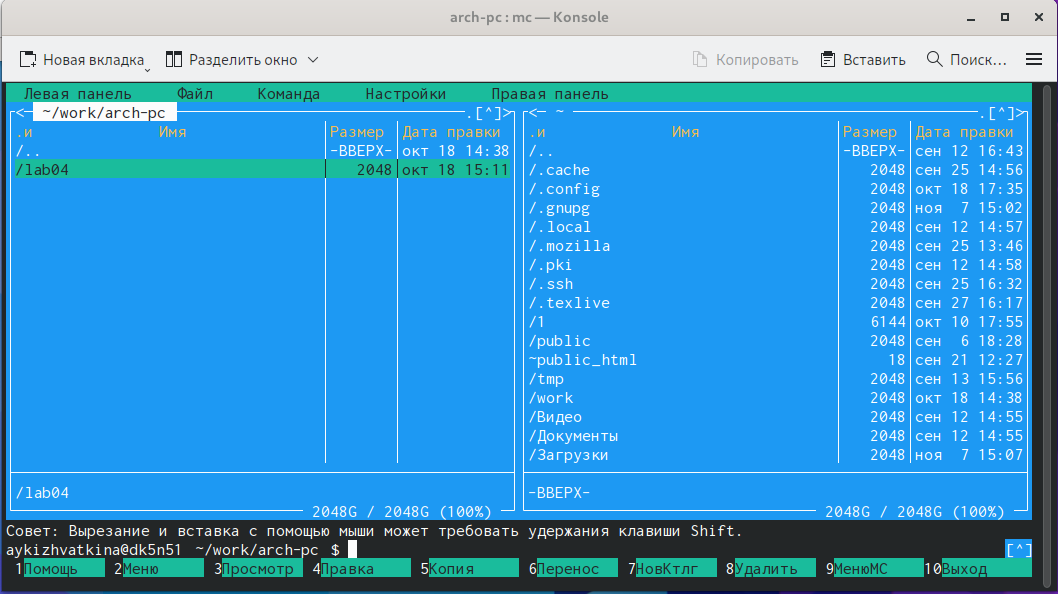


Рис. 2: Перемещение по каталогам.

С помощью функциональной клавиши F7 создаем папку lab05. (рис. 3)

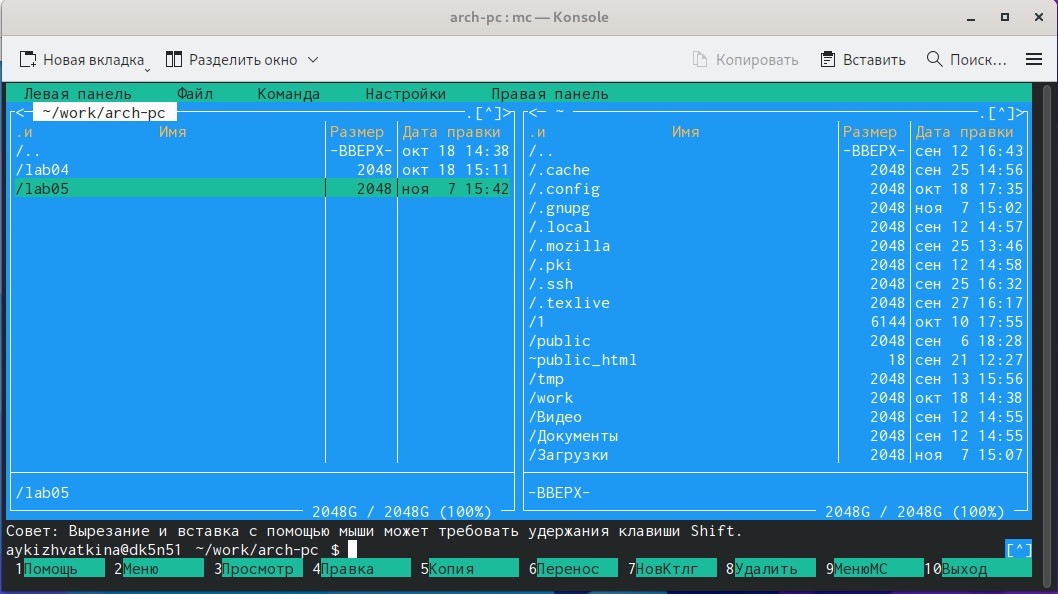


Рис. 3: Создание папки.

Переходим в созданный каталог. (рис. 4)

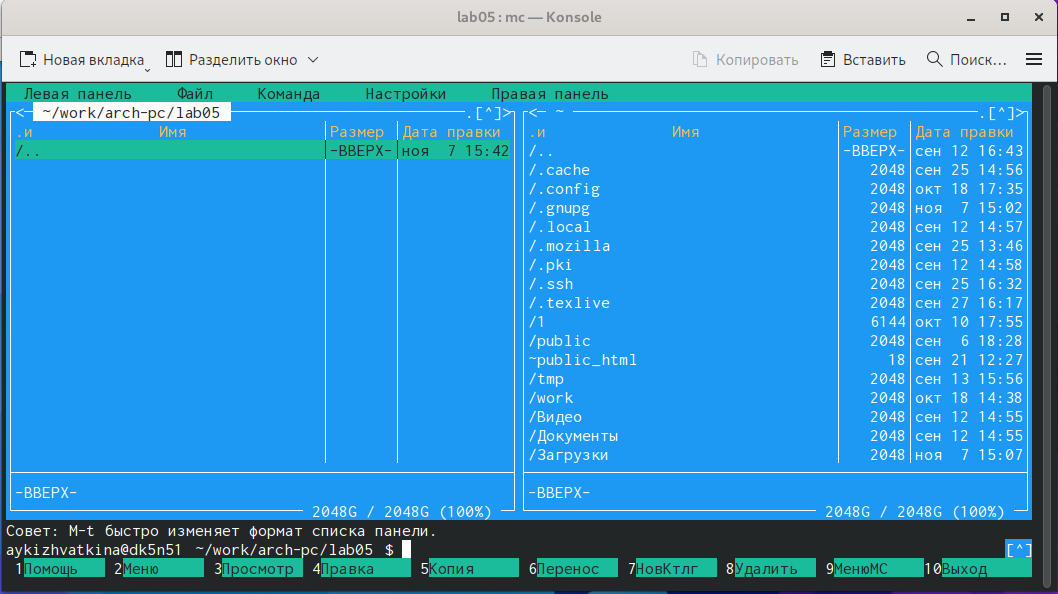


Рис. 4: Перемещение в каталог.

Пользуясь строкой ввода и командой touch создаем файл lab5-1.asm. (рис. 5)

Рис. 5: Создание файла.

Рис. 5: Создание файла.

С помощью функциональной клавши F4 открываем файл lab5-1.asm для редактирования во встроенном редакторе. Используем редактор nano или mcedit. Вводим текст программы из листинга 5.1. (рис. 6)

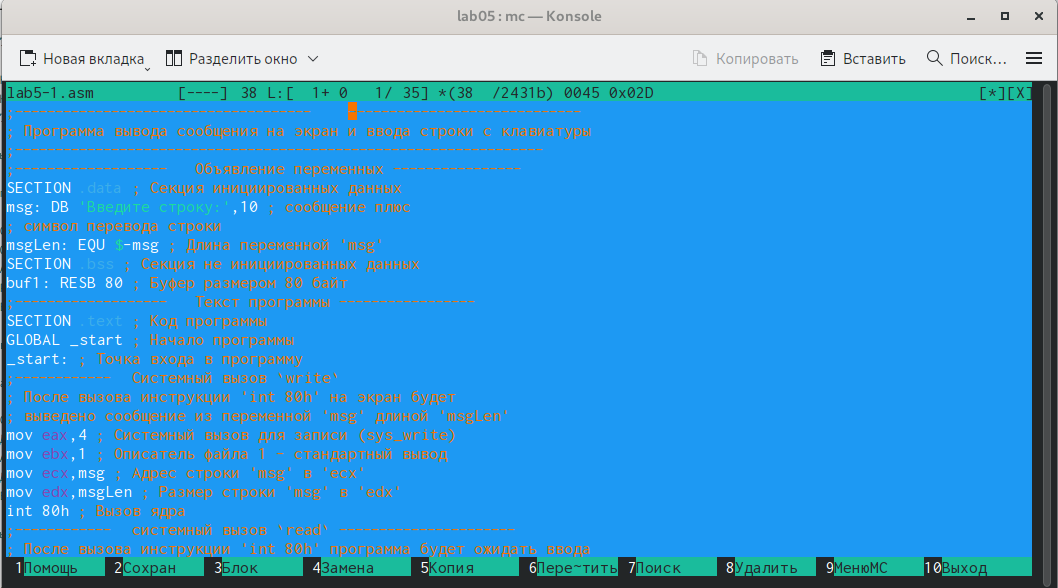


Рис. 6: Редактирование файла.

Сохраняем изменения и закрываем файл. (рис. 7)



Рис. 7: Сохранение изменений.

С помощью функциональной клавиши F3 откройте файл lab5-1.asm для просмотра. Проверяем, что файл содержит текст программы. (рис. 8)

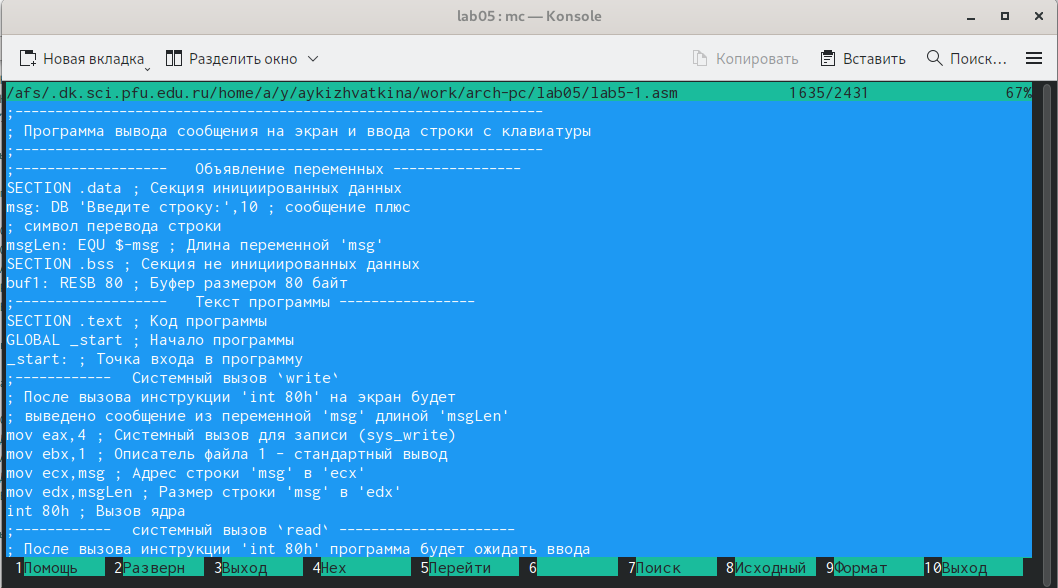


Рис. 8: Проверка файла.

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. Программа выводит строку ‘Введите строку:’ и ожидает ввода с клавиатуры. На запрос введите наше ФИО. (рис. 9)

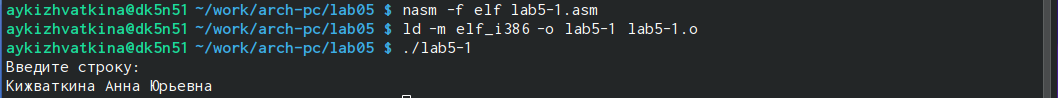


Рис. 9: Создание исполняемого файла. Проверка программы.

Скачиваем файл in\_out.asm. Копируем файл в тот же каталог, что и файл с программой. (рис. 10)

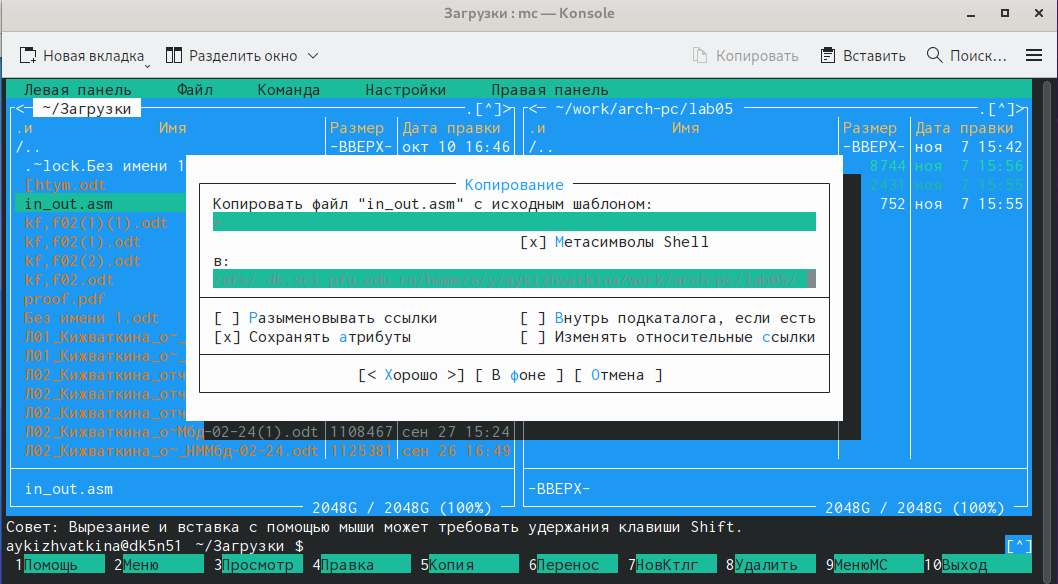


Рис. 10: Копирование файла в каталог.

С помощью функциональной клавиши F6 создаем копию файла lab5-1.asm с именем lab5-2.asm. (рис. 11)

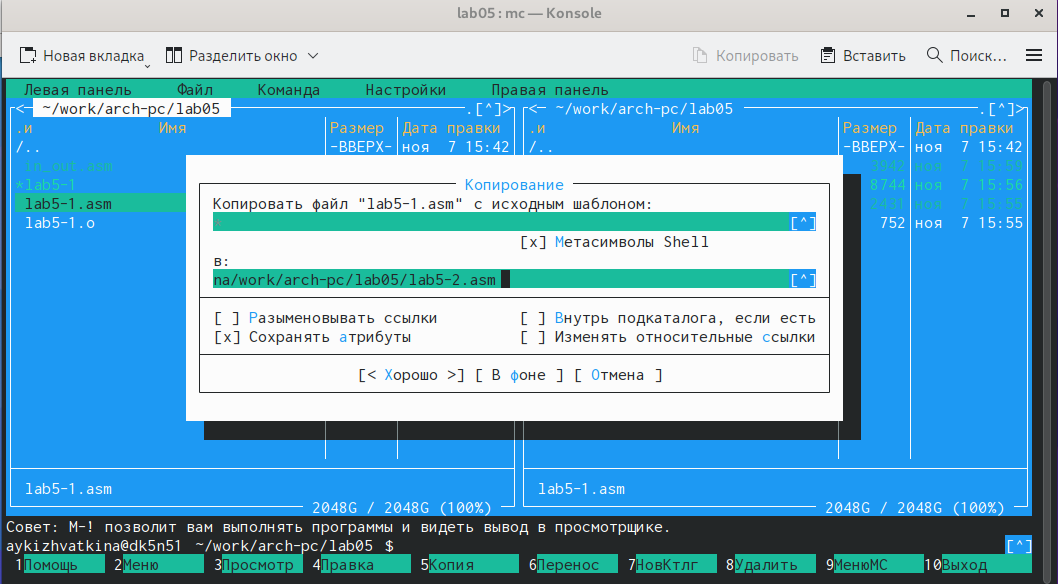


Рис. 11: Создание копии файла.

Исправляем текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm, используя подпрограммы sprintLF, sread и quit. (рис. 12)

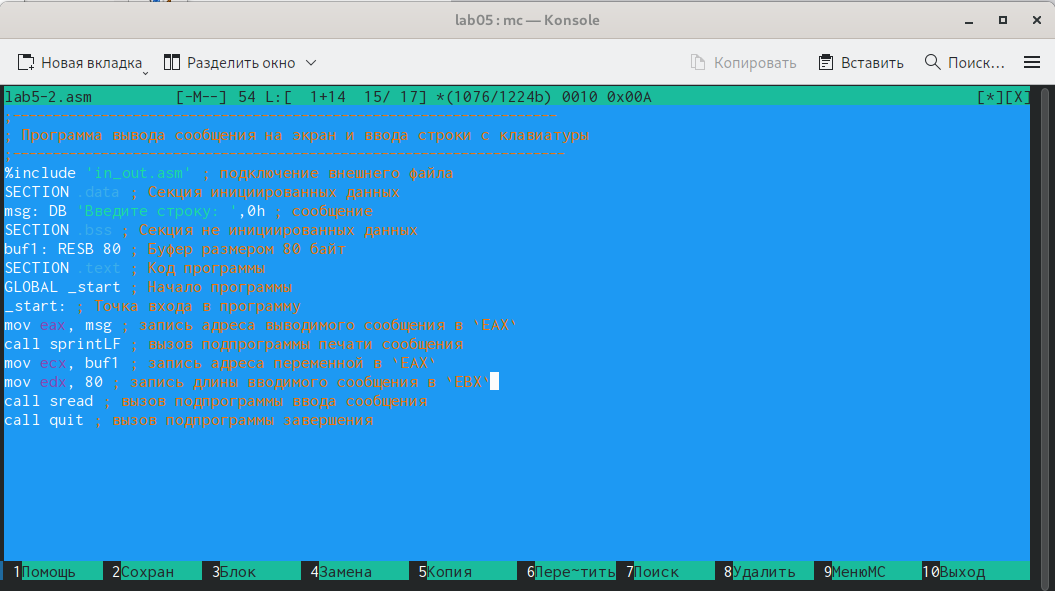


Рис. 12: Редактирование файла.

Создаем исполняемый файл и проверяем его работы. (рис. 13)

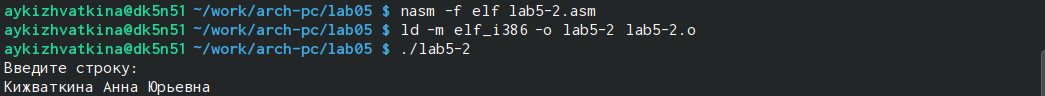


Рис. 13: Создание исполняемого файла. Проверка работы.

В файле lab5-2.asm заменяем подпрограмму sprintLF на sprint и сохраняем. (рис. 14)

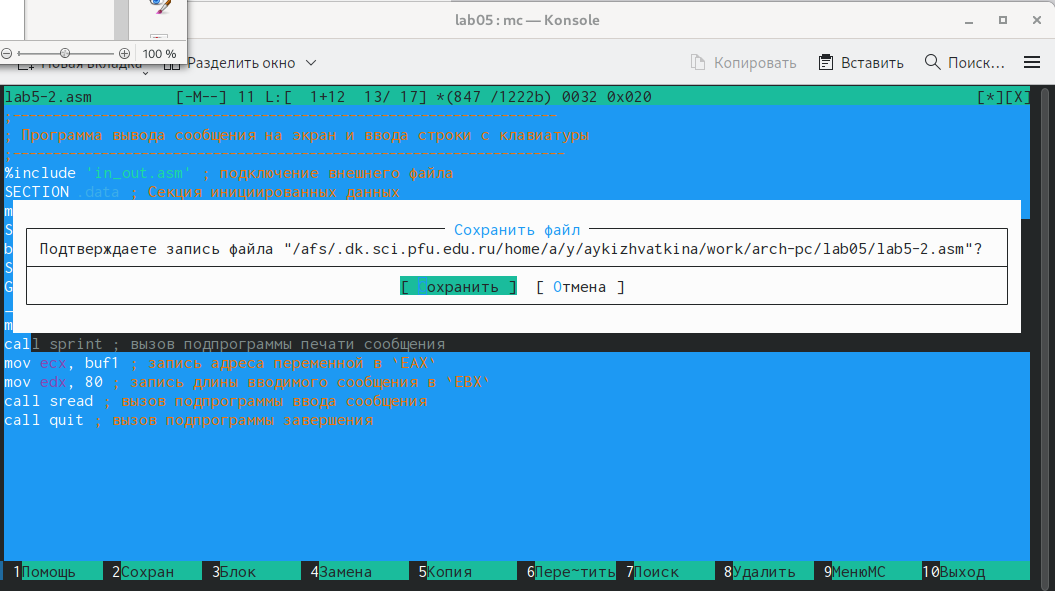


Рис. 14: Замена sprintLF на sprint. Сохранение файла.

Создаем файл и проверяем его работу. (рис. 15)

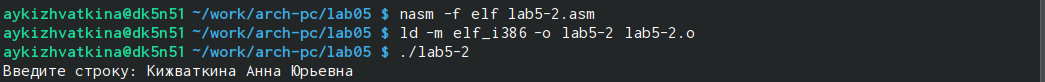


Рис. 15: Создание исполняемого файла. Проверка работы.

Разница в том что sprintLF переносит на новую строку, а с sprint нужно вводить на той же строке.

1. Выполнение самостоятельной работы.

Создаем копию файла lab5-1.asm. (рис. 16)

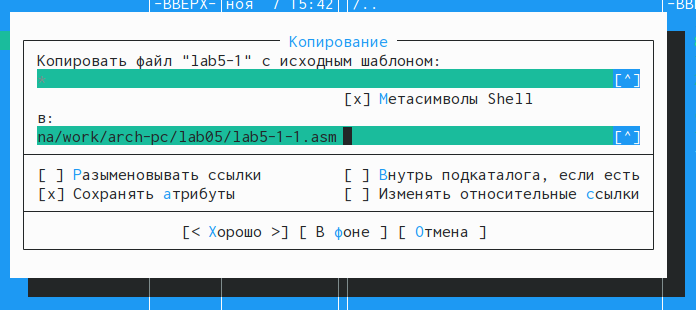


Рис. 16: Создание копии.

Вносим изменения в программу(без использования внешнего файла in\_out.asm) (рис. 17)

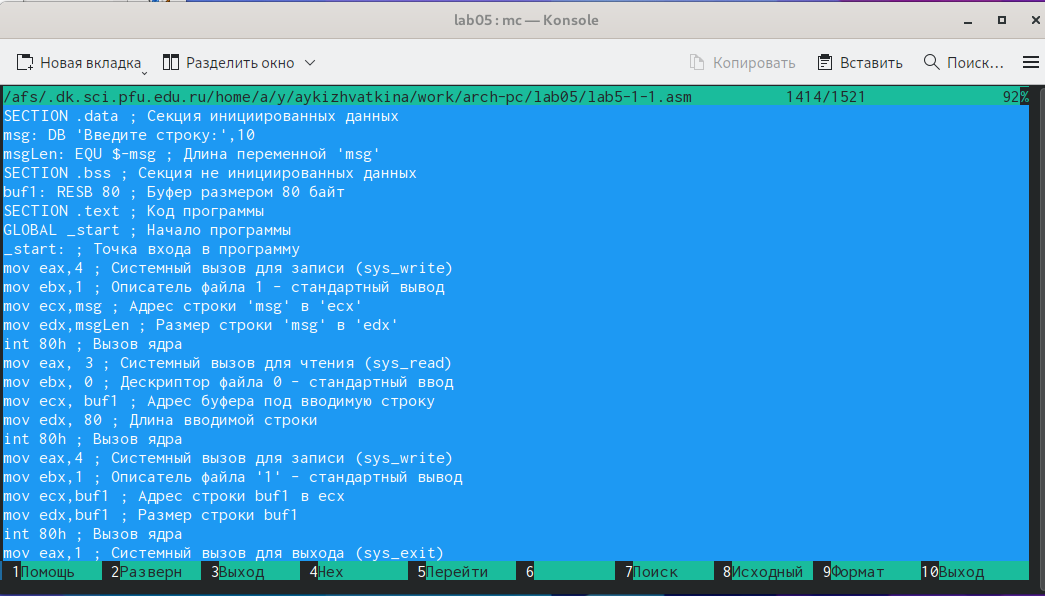


Рис. 17: Редактирование файла.

Получаем исполнительный файл. Проверяем правильность выполнения. На приглашение ввести строку вводим свою фамилию. (рис. 18)

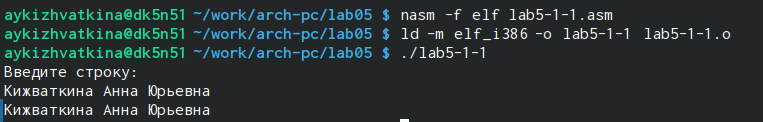


Рис. 18: Создание исполнительного файла. Проверка выполнения.

Создаем копию файла lab5-2.asm. (рис. 19)

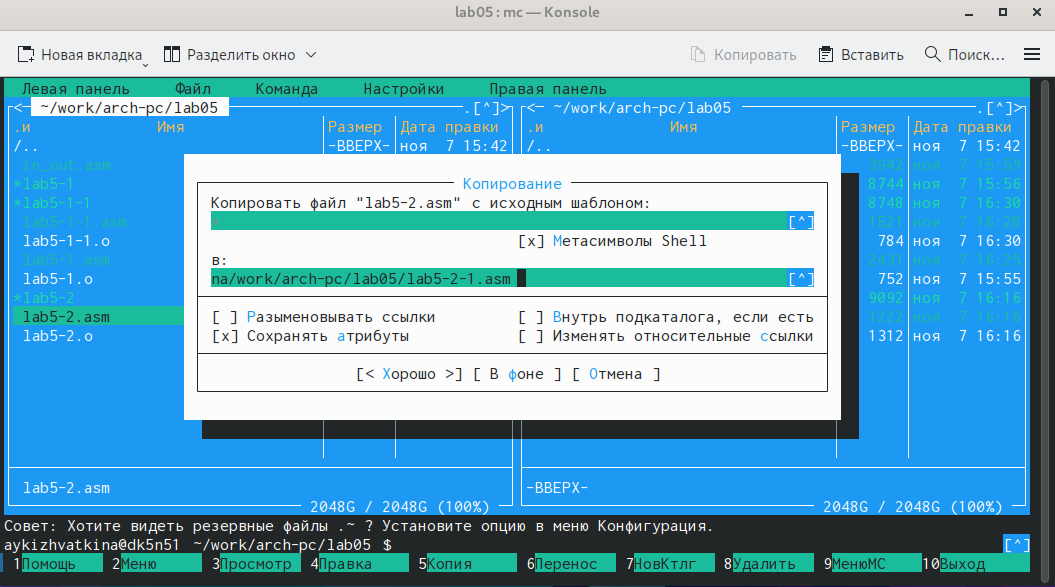


Рис. 19: Создание копии файла.

Исправляем текст программы с использованием программ из внешнего файла in\_out.asm. (рис. 20)

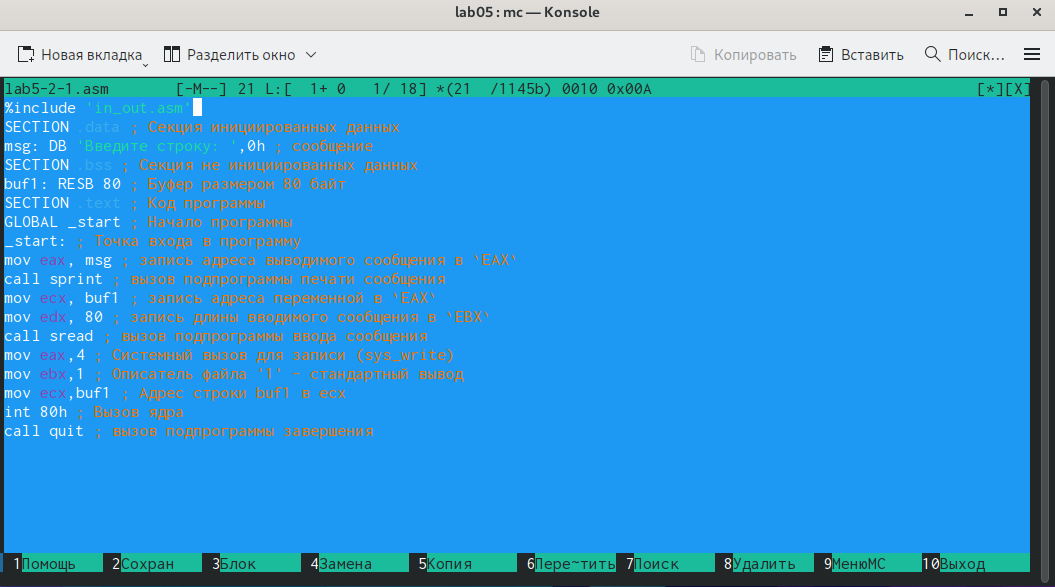


Рис. 20: Редактирование файла.

Создаем исполняемый файл и проверяем его работу. (рис. 21)

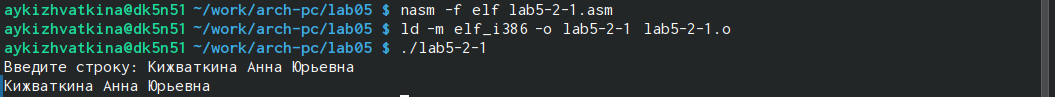


Рис. 21: Создание исполняемого файла. Проверка работы.

# 5 Выводы

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