Лабораторная работа №5

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

Бунин Арсений Викторович

Содержание

# 1 Цель работы

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

# 2 Задание

1. Создать папку и файл на языке Ассемблер в ней с помощью Midnight Commander
2. Создать файл на языке Ассемблер, принимающий на вход строку
3. Загрузить файлы на github

# 3 Теоретическое введение

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. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter (рис. 5.1). В Midnight Commander используются функциональные клавиши F1 — F10 , к которым привязаны часто выполняемые операции

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).

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. Инструкция языка ассемблера int предназначена для вызова прерывания с указанным номером.

# 4 Выполнение лабораторной работы

Открываем терминал и запускаем Midnight Commander(рис. [1](#fig:fig1)).

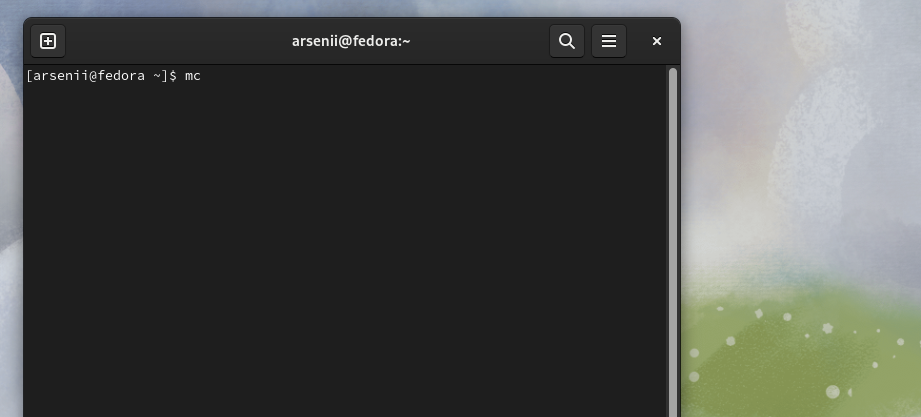


Figure 1: Вызов Midnight Commander

Переходим в каталог курса (рис. [2](#fig:fig2)).

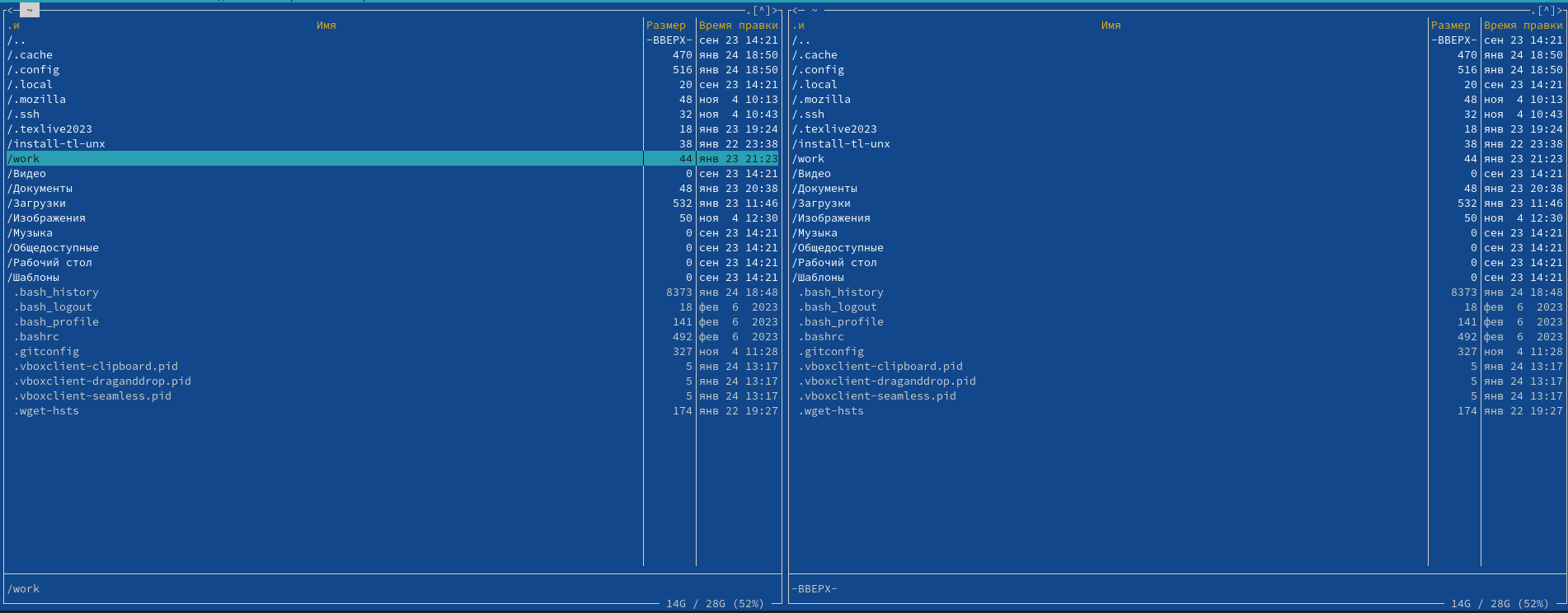


Figure 2: Каталог курса в Midnight Commander

Создаем папку для пятой лабораторной работы (рис. [3](#fig:fig3)).

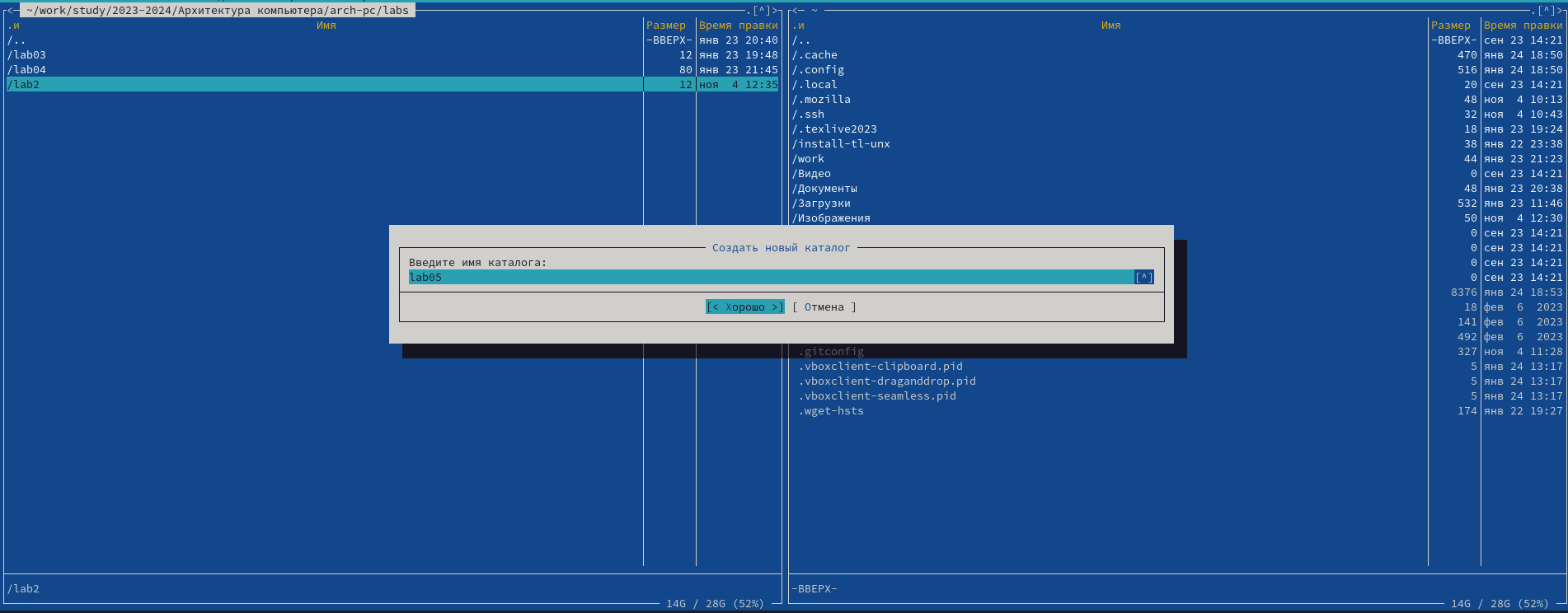


Figure 3: Окно создания папки

Создаем файл lab5-1.asm в новой папке (рис. [4](#fig:fig4)).

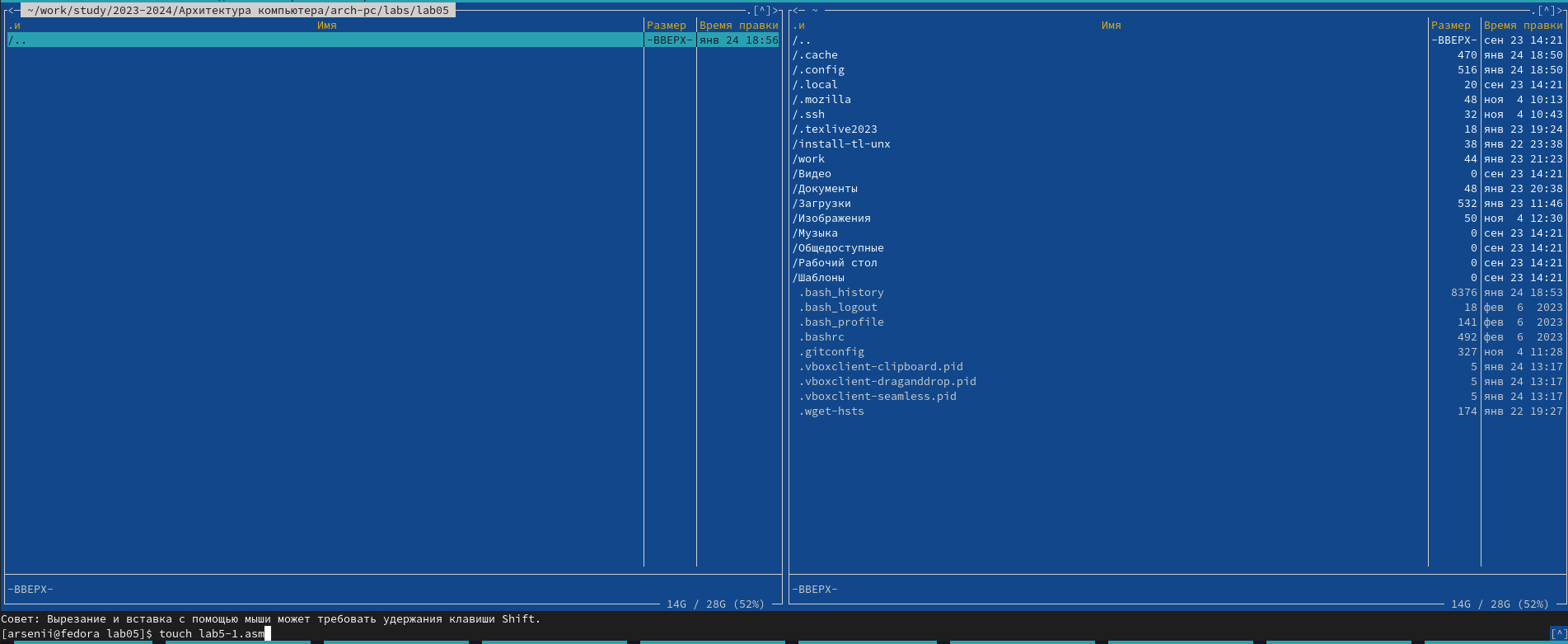


Figure 4: Окно создания файла

Записываем код в файл в редакторе mcedit (рис. [5](#fig:fig5))

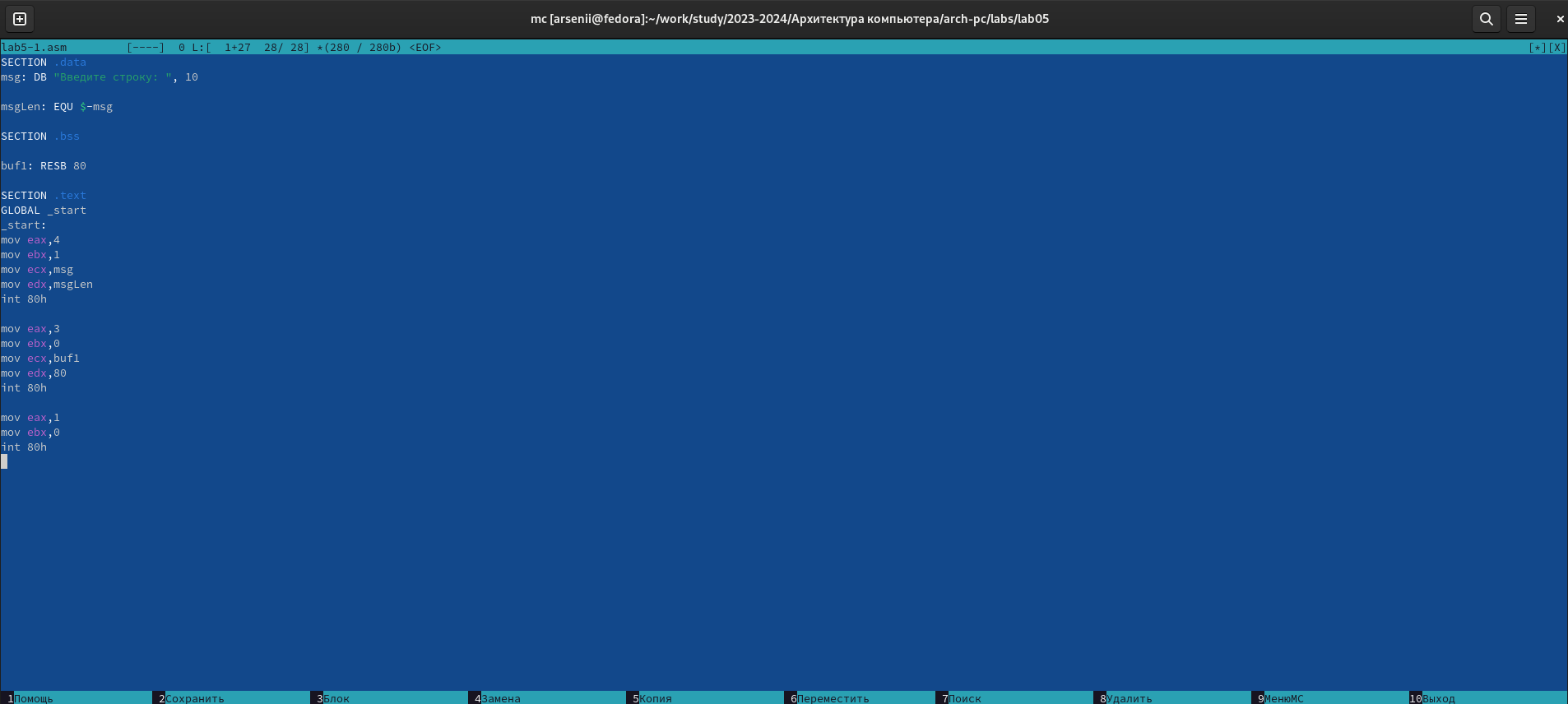


Figure 5: Файл в редакторе mcedit

Проверяем файл в режиме просмотра (рис. [6](#fig:fig6))

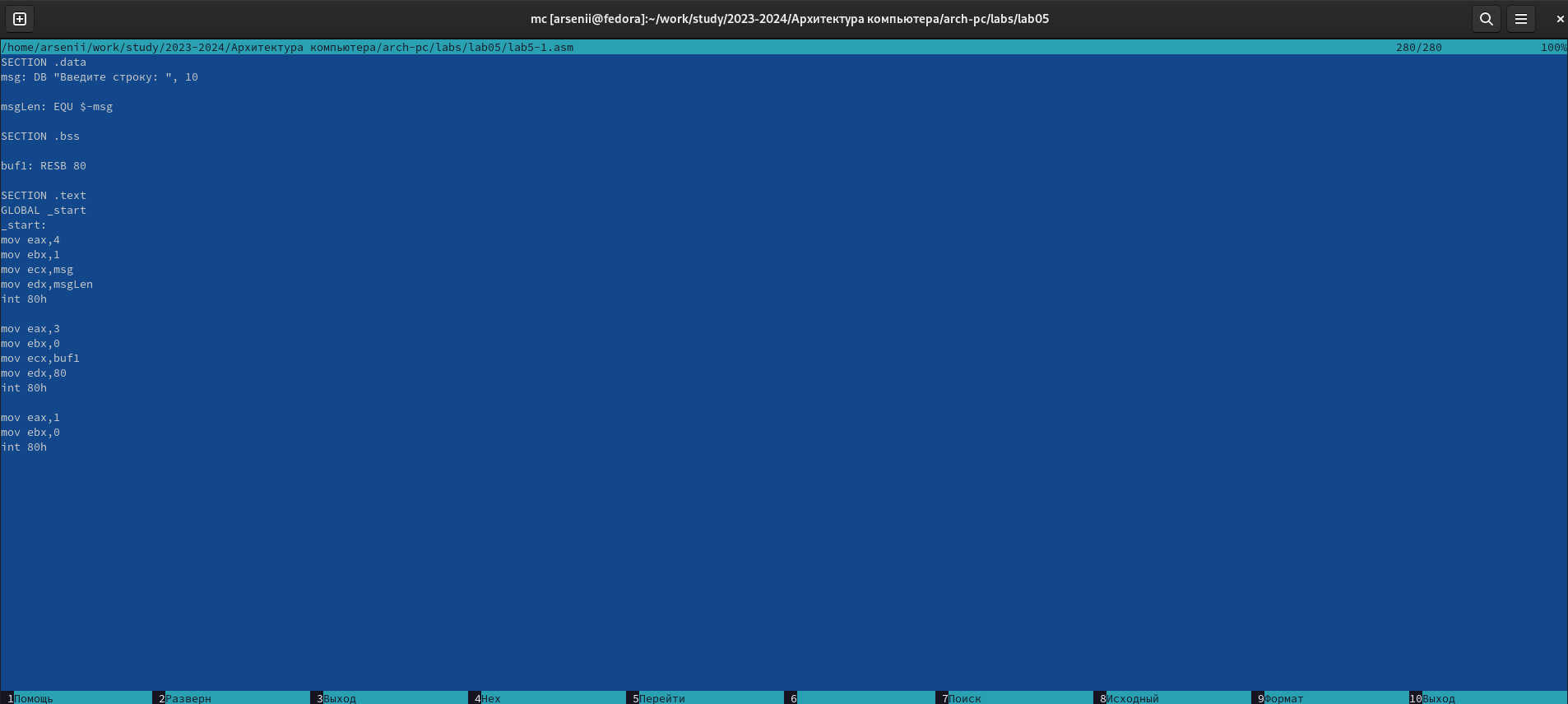


Figure 6: Файл в режиме просмотра

Выполняем линковку и компоновку (рис. [7](#fig:fig7))

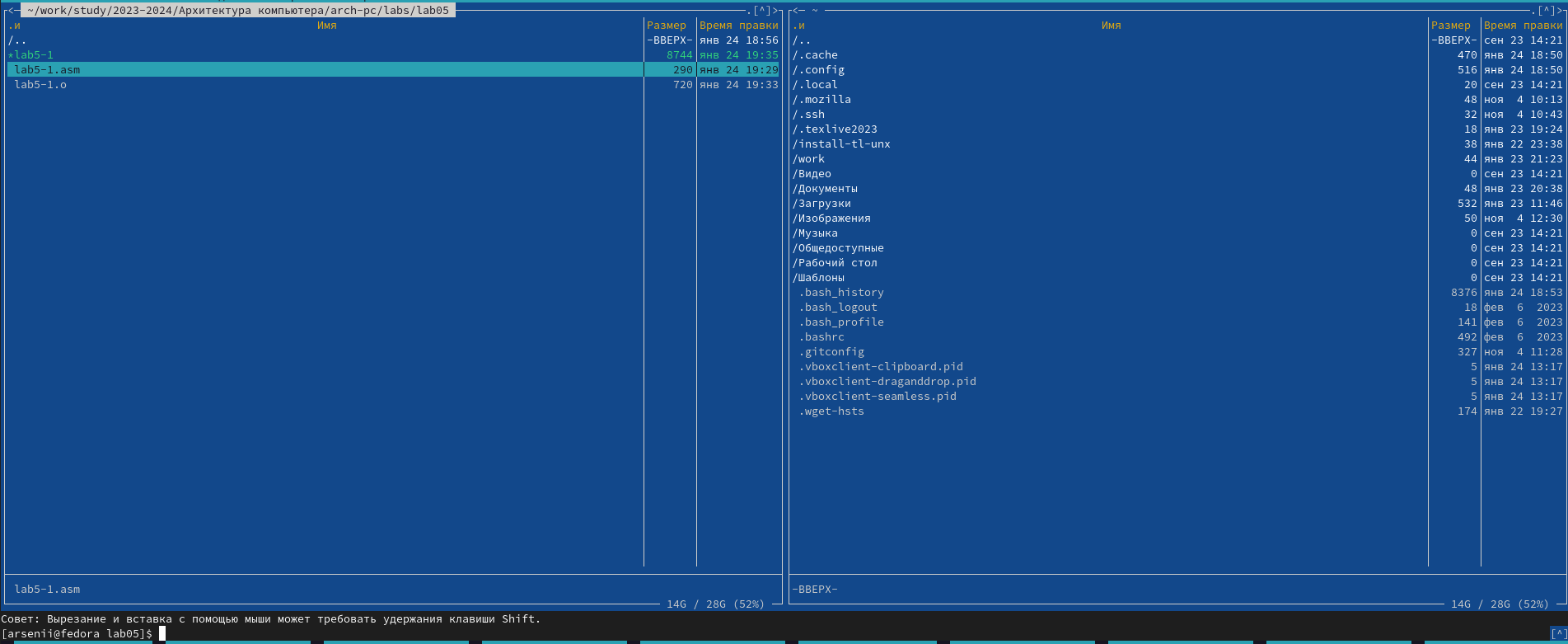


Figure 7: Линковка и компоновка

Результат работы программы(рис. [8](#fig:fig8))

Figure 8: Работа первой программы

Figure 8: Работа первой программы

Переименовываем файл в lab5-1.asm(рис. [9](#fig:fig9))

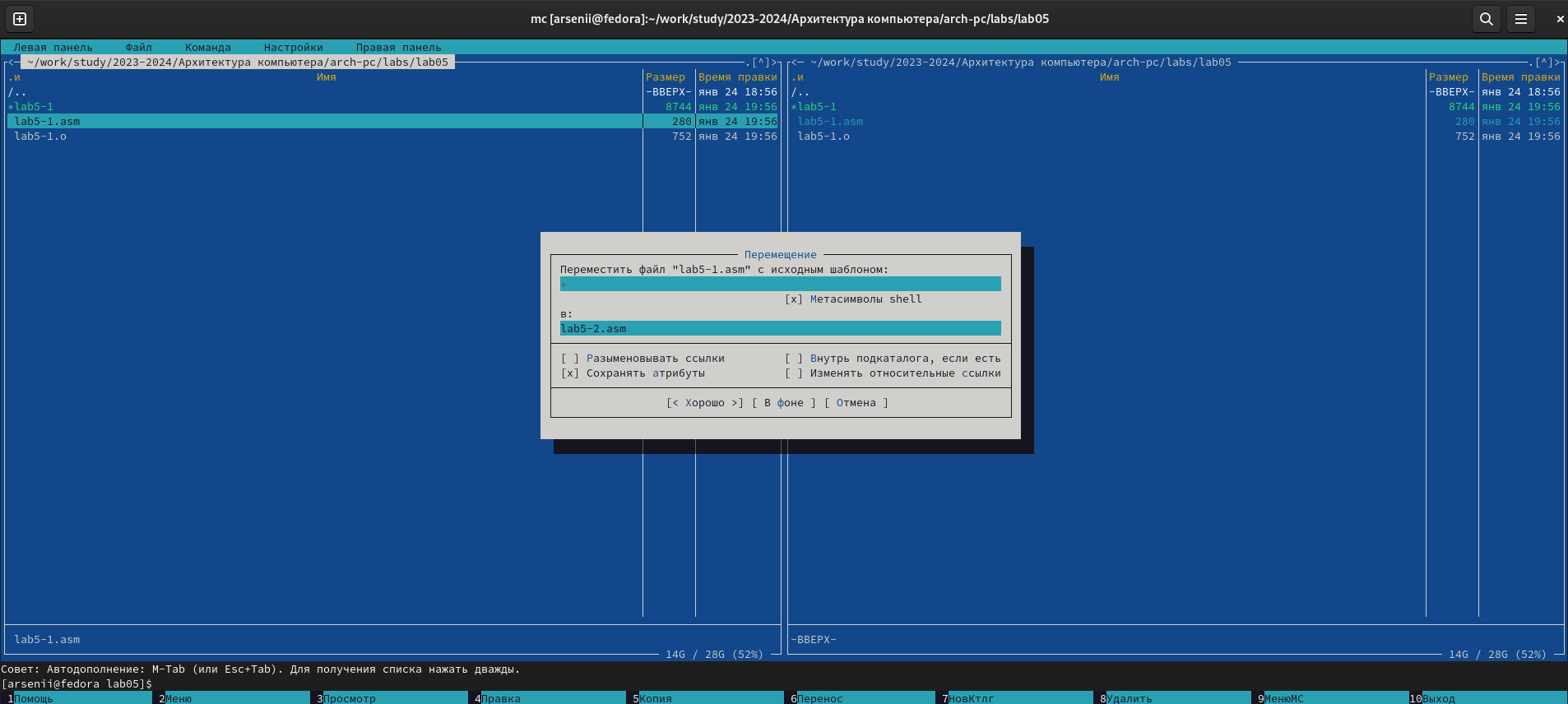


Figure 9: Окно замены файла

Записываем код второй программы в файл(рис. [10](#fig:fig10))

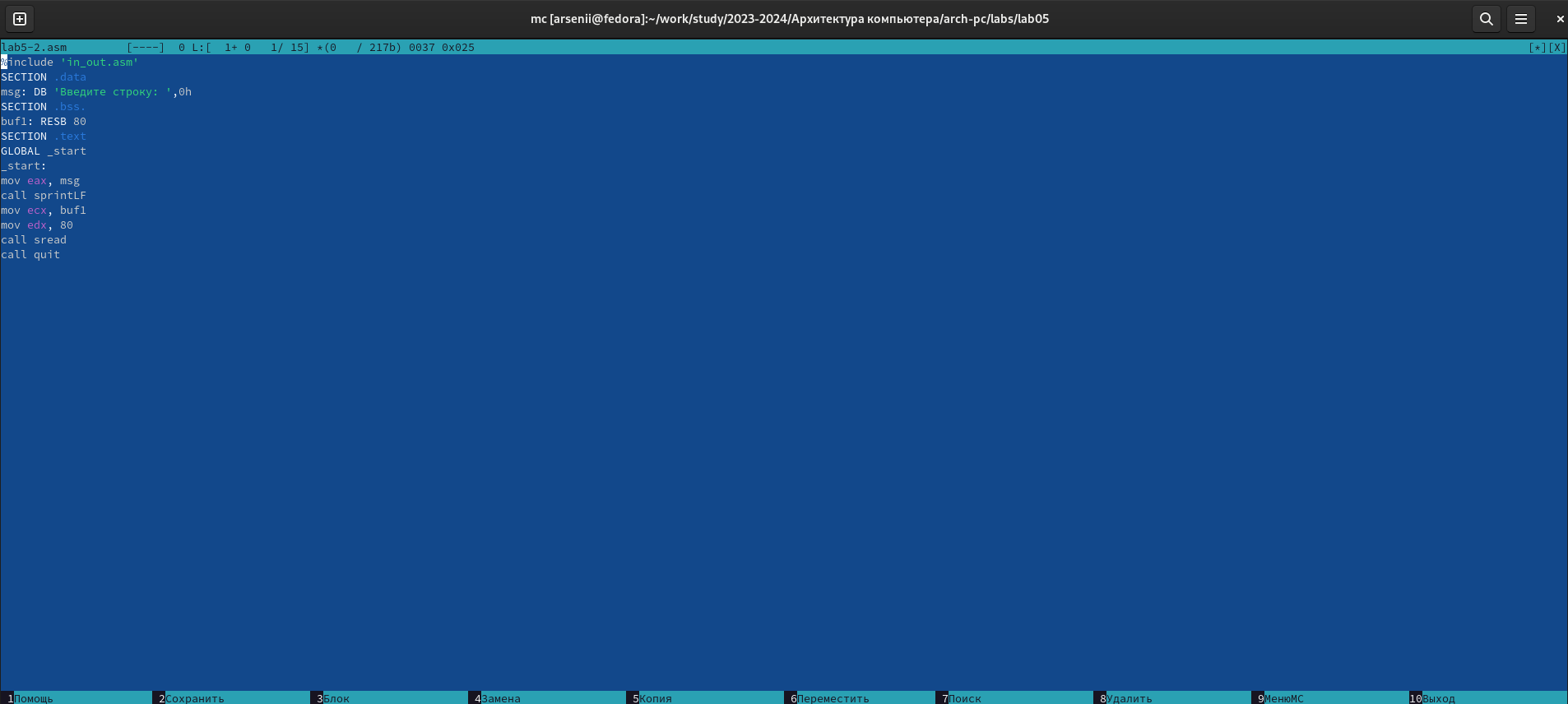


Figure 10: Код второй программы

Результат работы второй программы(рис. [11](#fig:fig11))

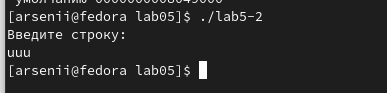


Figure 11: Вторая программа

Результат работы измененной второй программы. Теперь ввод текста происходит на строке с текстовым сообщением, а не на новой строке(рис. [12](#fig:fig12))

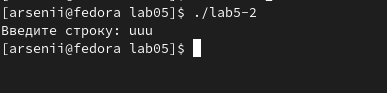


Figure 12: Измененная вторая программа

# 5 Самостоятельная работа

1. Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.

Текст третьей программы (рис. [13](#fig:fig15))



Figure 13: Третья программа

Результат работы третьей программы (рис. [14](#fig:fig13))

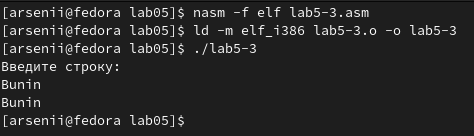


Figure 14: Третья программа

1. Создайте копию файла lab5-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран. Не забудьте, подключаемый файл in\_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.
2. Создайте исполняемый файл и проверьте его работу.

Текст четвертой программы (рис. [15](#fig:fig16))

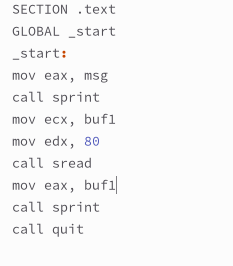


Figure 15: Третья программа

Результат работы четвертой программы (рис. [16](#fig:fig14))

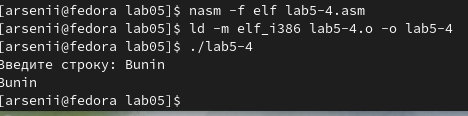


Figure 16: Четвертая программа

# 6 Выводы

Освоили работу с Midnight Commander и научились использовать команды int и mov языка Assembler

# Список литературы