Лабораторная работа номер 5

Архитектура компьютера

Титков Ярослав Максимович

Содержание

# 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 программ
3. Подключение файлов
4. Фиксация всей работы
5. Самостоятельная работа

# 3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управ- лению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Про- грамма 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 разме- ром 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 (учетве- рённое слово); - DT (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых перемен- ных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. mov dst,src Здесь операнд dst — приёмник, а src — источ- ник. В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const). Инструкция языка ассемблера intпредназначена для вызова прерывания с указанным номером. int n Здесь n — номер прерывания, принадлежащий диапазону 0–255. При програм- мирова- нии в Linux с использованием вызовов ядра sys\_calls n=80h (принято задавать в шестнадцатеричной системе счисления)

# 4 Выполнение лабораторной работы

1. Основы работы с Midnight Commander

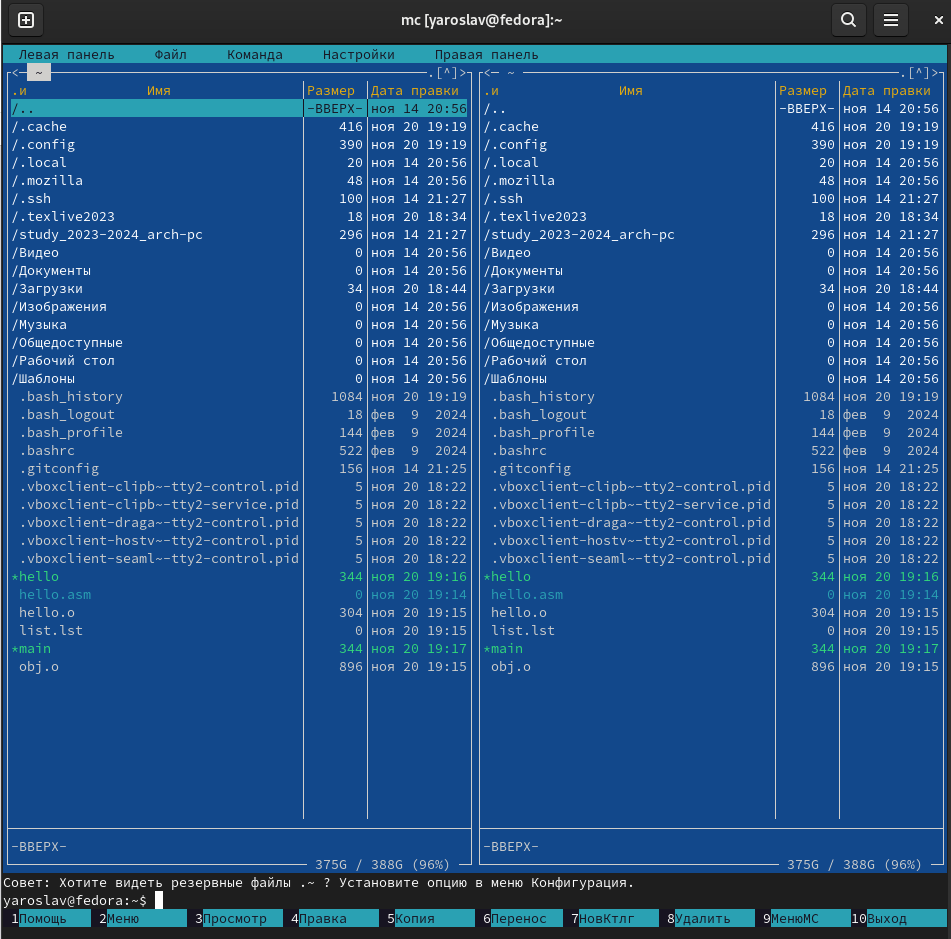


Рис. 1: Пользуясь командой mc открыл Midnight Commander

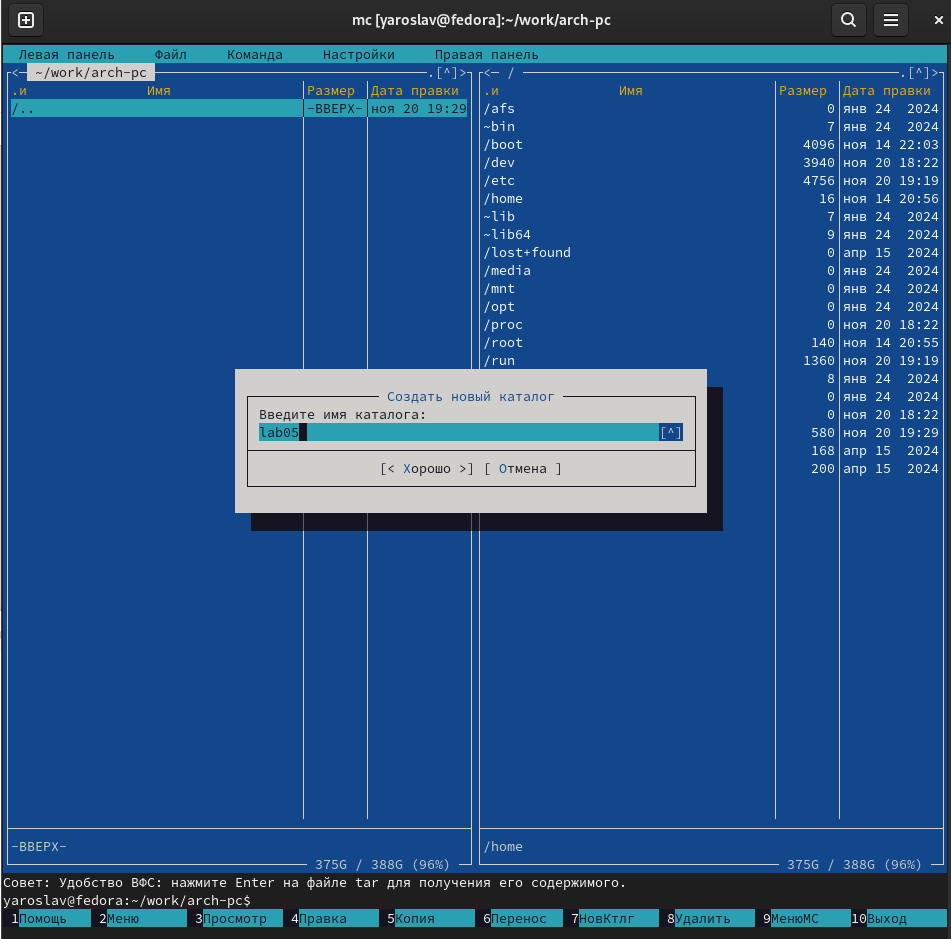


Рис. 2: Создал папку lab2

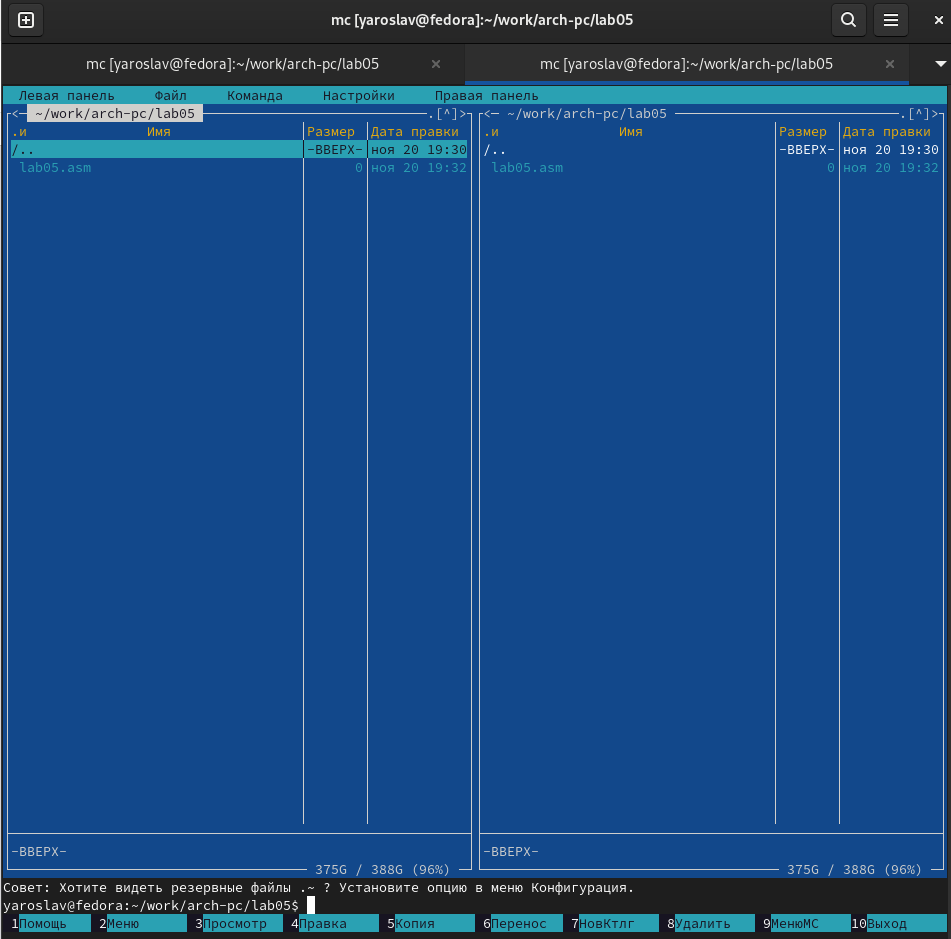


Рис. 3: С помощью команды touch создал asm файл и проверил в Midnight Commander

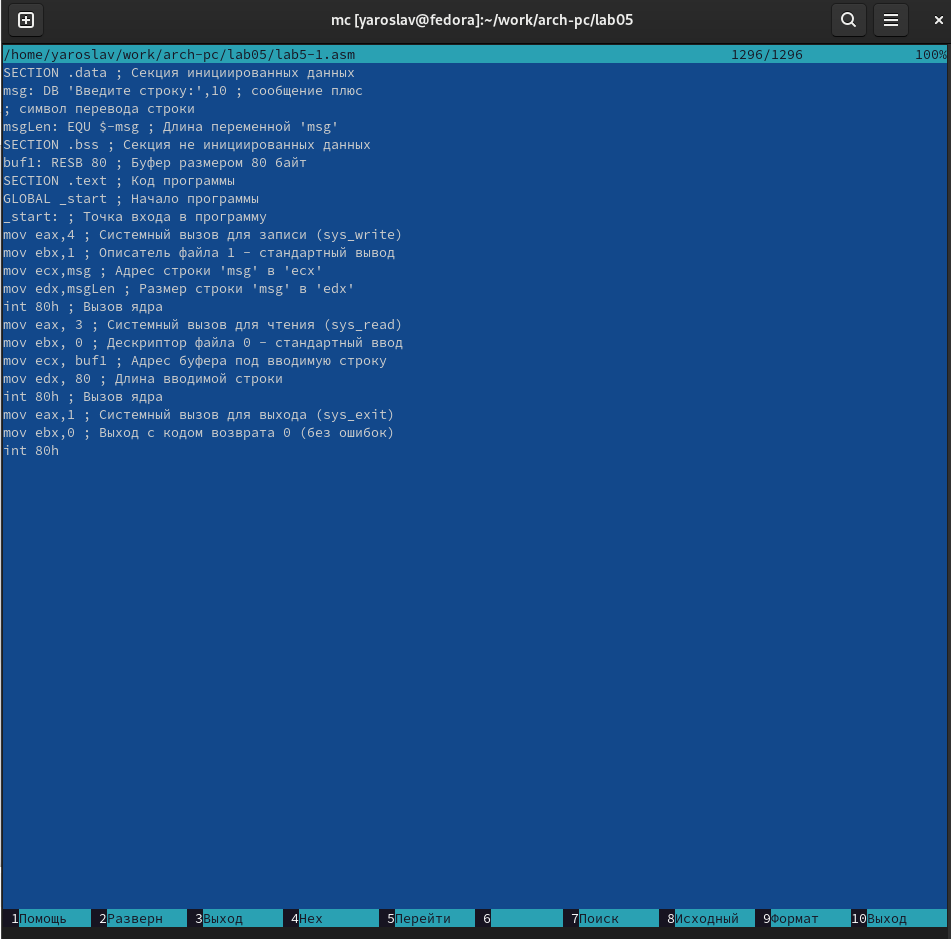


Рис. 4: С помощью клавиши F4 открыл файл lab5-1.asm и ввёл предложенный текст

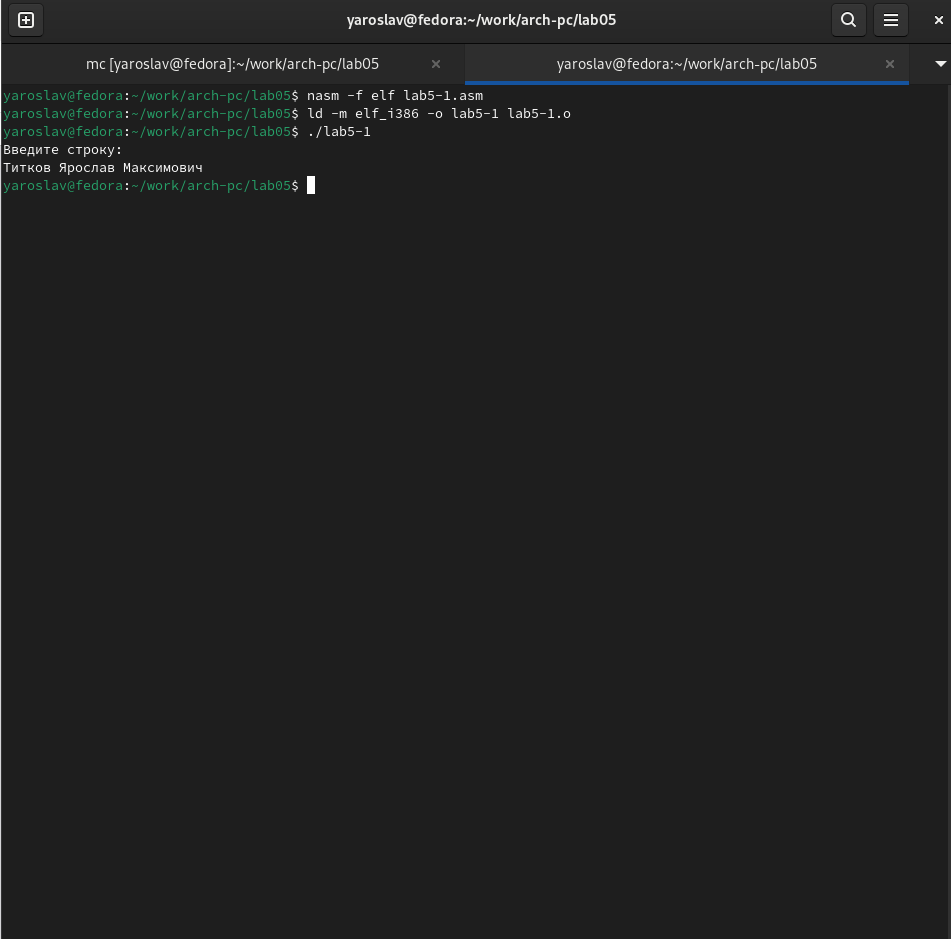


Рис. 5: Прописываю команды NASM и запускаю файл

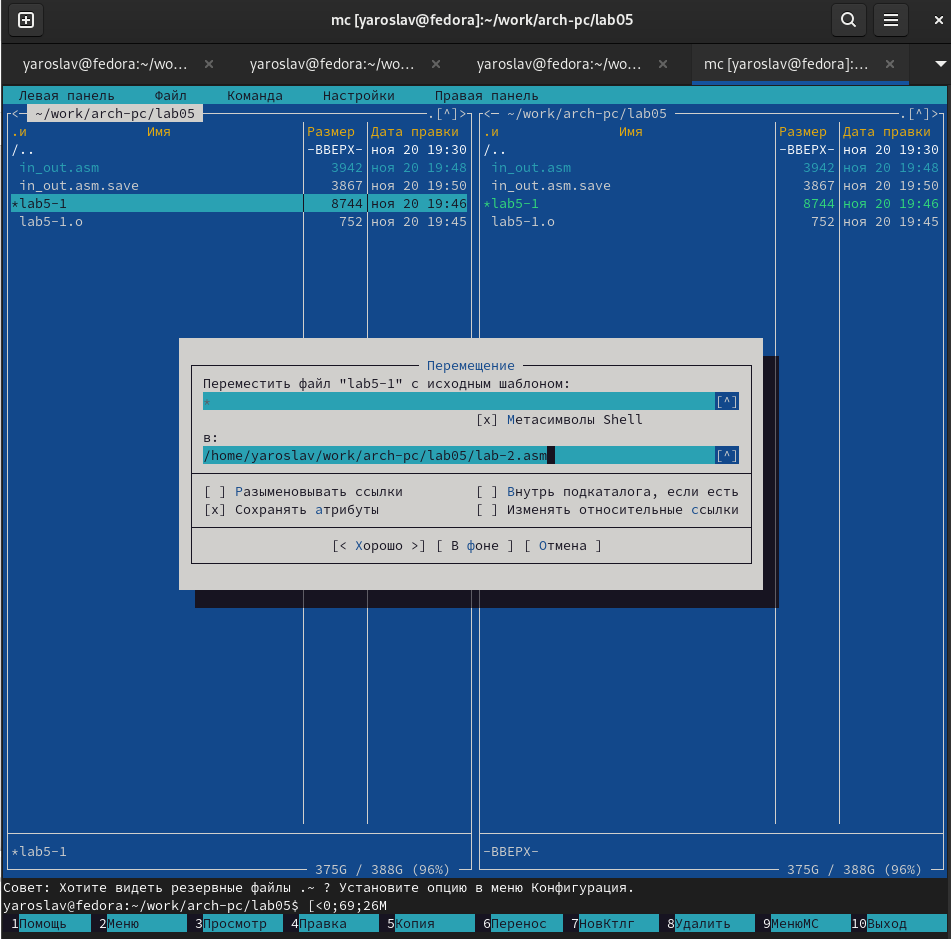


Рис. 6: C помощью клавиши F6 создал копию файла lab5-1.asm с именем lab5-2.asm

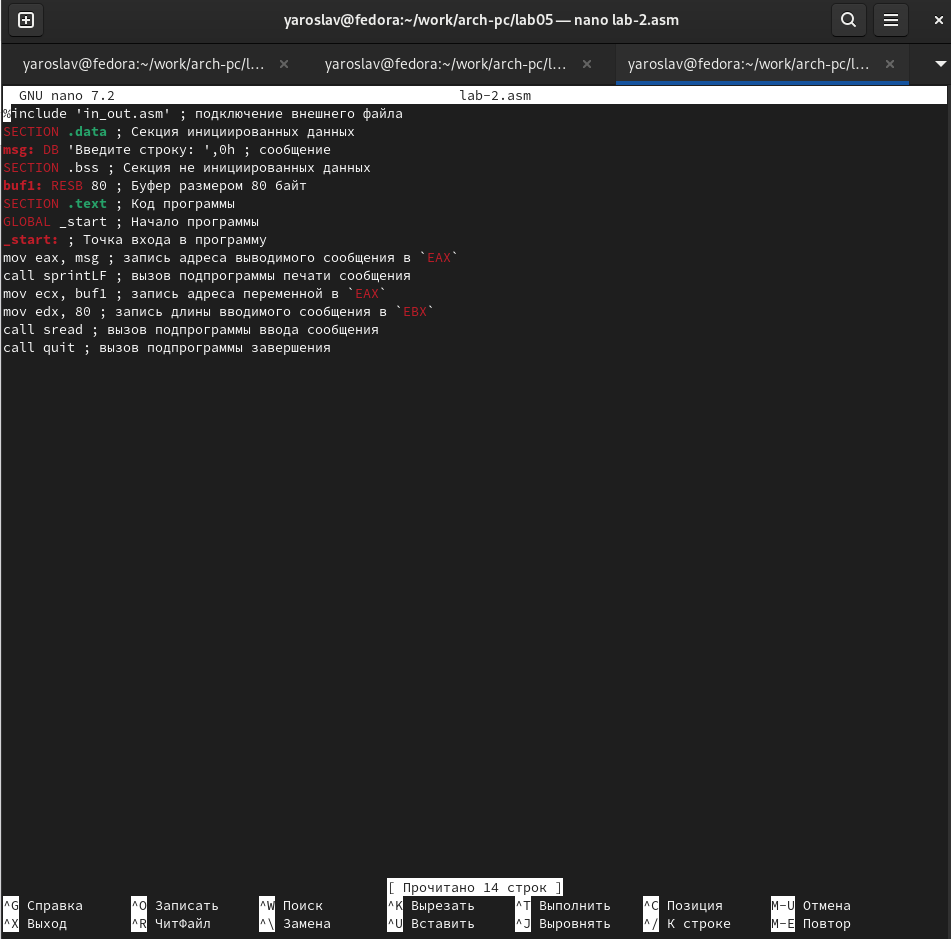


Рис. 7: С помощью команды nano исправил текст в новом файле

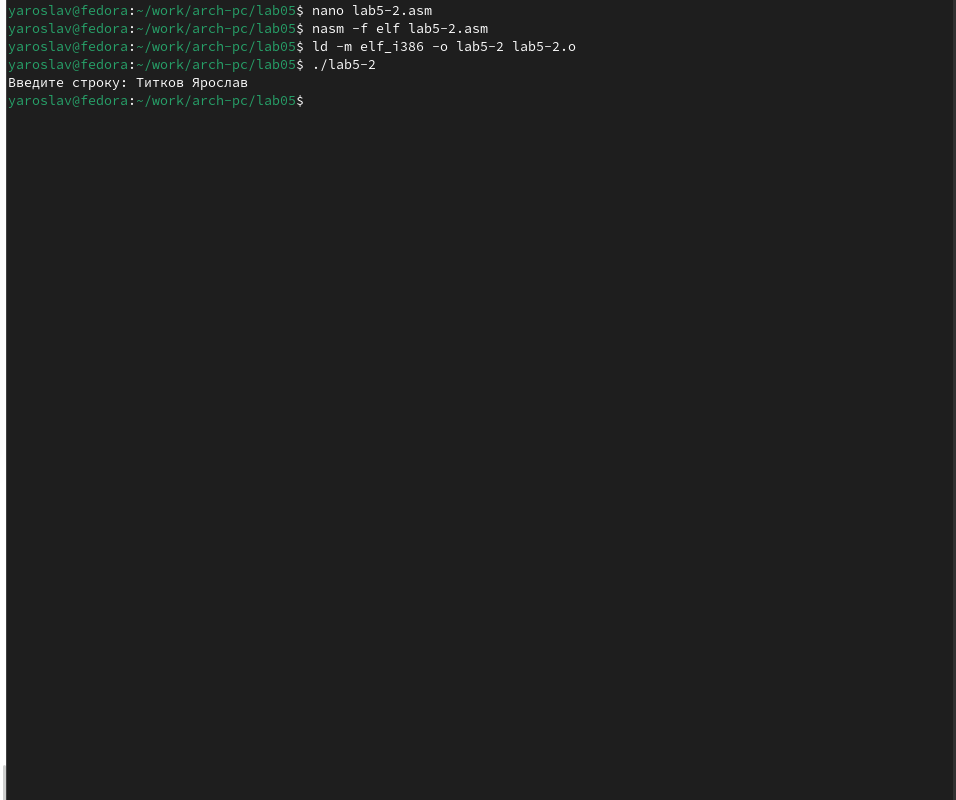


Рис. 8: Отредактировал текст так, чтобы строчки были на одном уровне и запустил файл

1. Задания для самостоятельной работы:
2. Создание копии файла lab5-1.asm и изменение программы

Я создал копию файла lab5-1.asm и назвал её lab5-1-modified.asm. Затем я внес изменения в программу, чтобы она работала по следующему алгоритму:

Вывести приглашение типа "Введите строку:".  
  
Ввести строку с клавиатуры.  
  
Вывести введенную строку на экран.

1. Получение исполняемого файла и проверка работы

Я получил исполняемый файл и проверил его работу. На приглашение ввести строку я ввел свою фамилию. 3. Создание копии файла lab5-2.asm и исправление программы

Я создал копию файла lab5-2.asm и назвал её lab5-2-modified.asm. Затем я исправил текст программы с использованием подпрограмм из внешнего файла in\_out.asm, чтобы она работала по следующему алгоритму:

Вывести приглашение типа "Введите строку:".  
  
Ввести строку с клавиатуры.  
  
Вывести введенную строку на экран.

1. Создание исполняемого файла и проверка работы

Я создал исполняемый файл и проверил его работу.

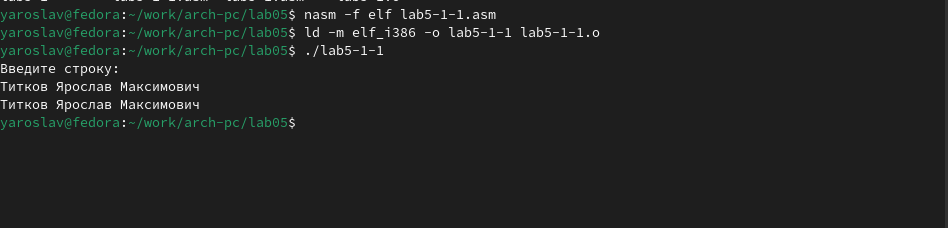


Рис. 9: Скопировал файл lab5-1.asm и задал ему имя lab5-1-1.asm, а затем с помощью nano изменил алгоритм на нужный для выполнения самостоятельной работы

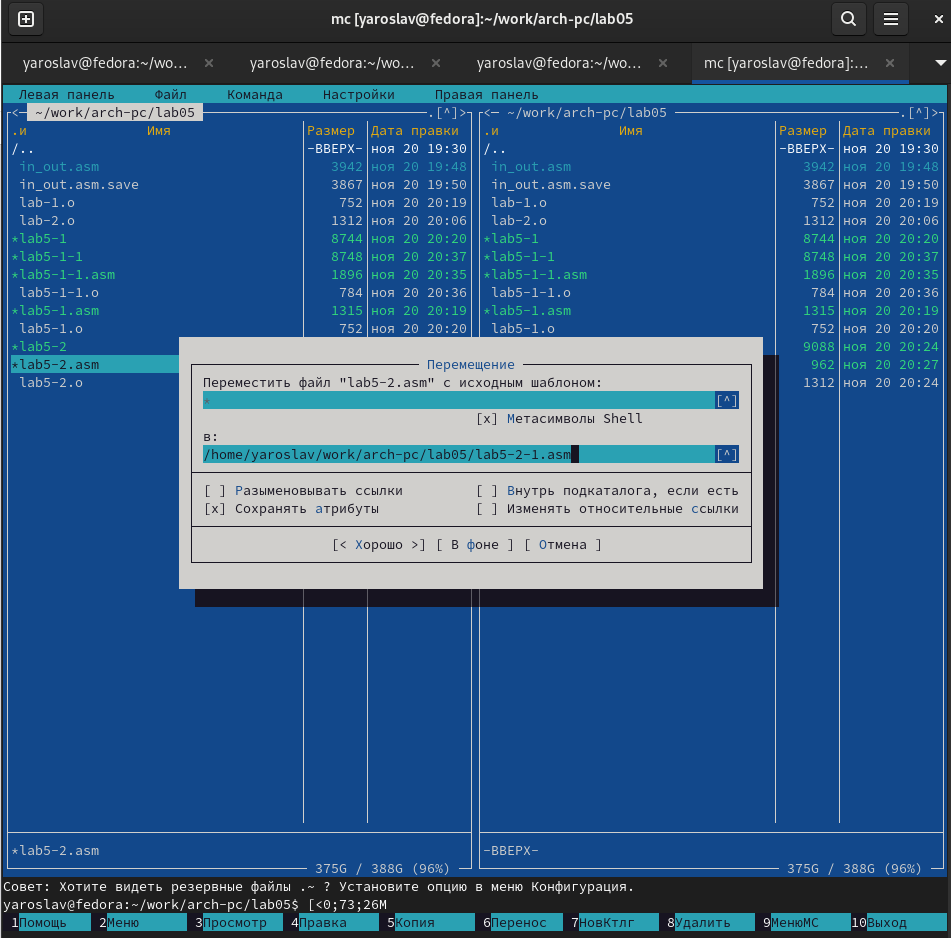


Рис. 10: Проделал тоже самое с файлом lab5-2

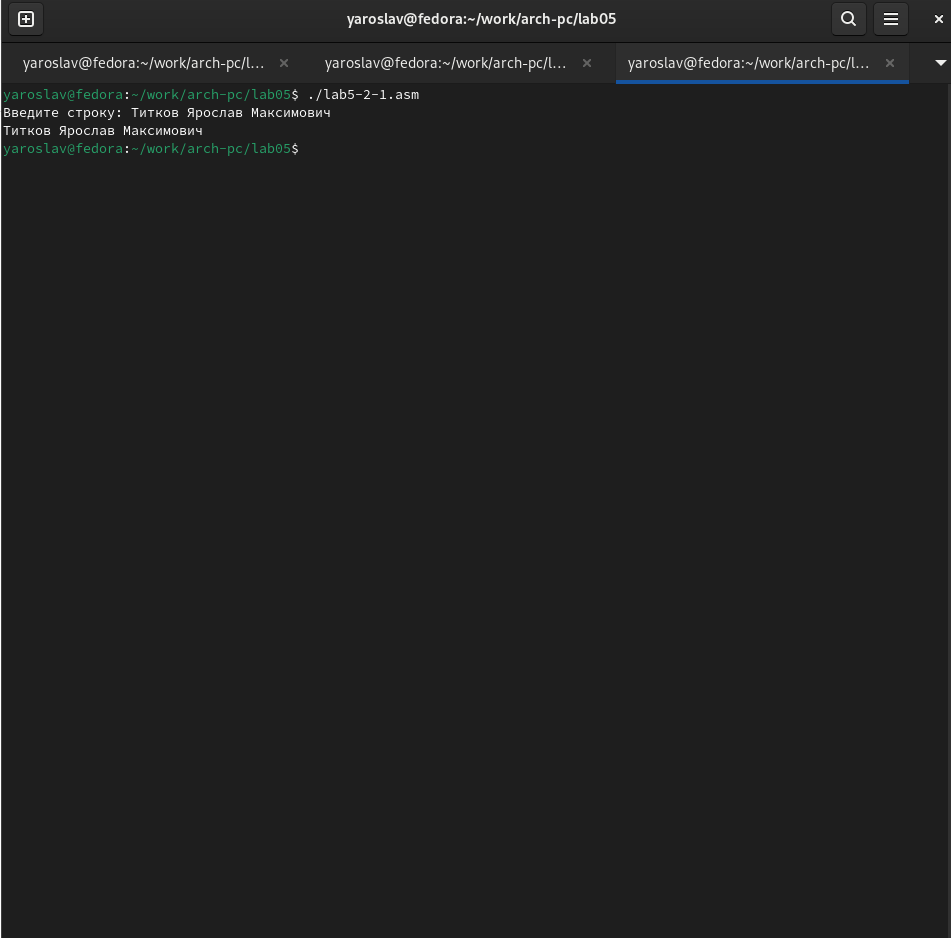


Рис. 11: запустил файл lab5-2-2

# 5 Выводы:

При выполнении данной лабораторной работы я приобрел практические навы- ки работы в Midnight Commander, а также освоил инструкции языка ассемблера mov и int.