Шаблон отчёта по лабораторной работе

Простейший вариант

Дмитрий Сергеевич Кулябов

Содержание

# 1 Цель работы

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

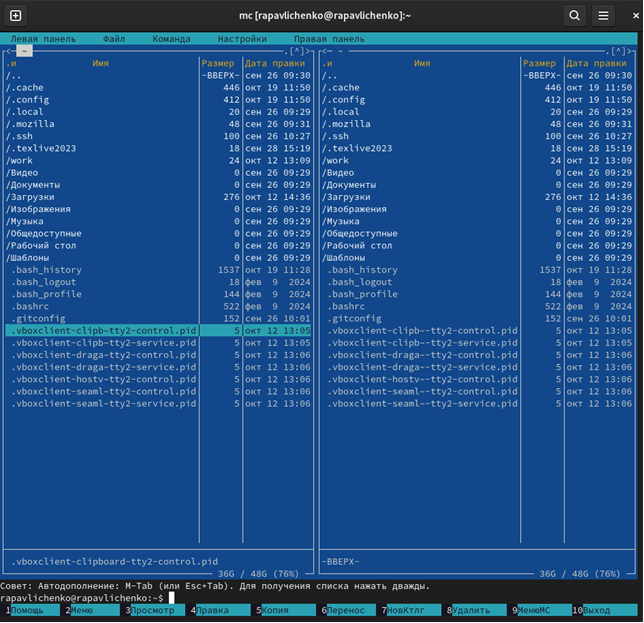
# 2 Выполнение лабораторной работы

Для начала выполнения лабораторной работы нам необходимо открыть Midnight commander с помощью команды mc

Запуск Midnight commander

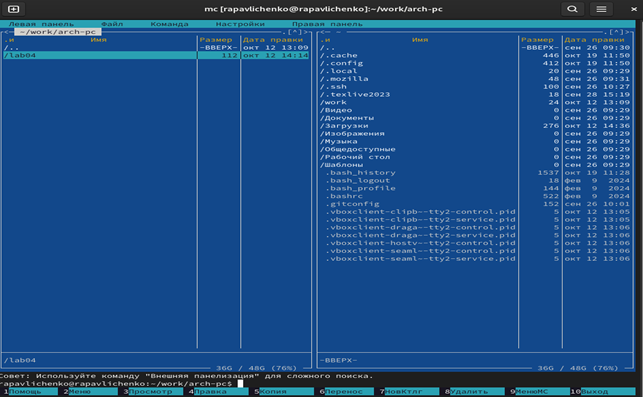
Запуск Midnight commander

После ввода команды мы увидим такой интерфейс



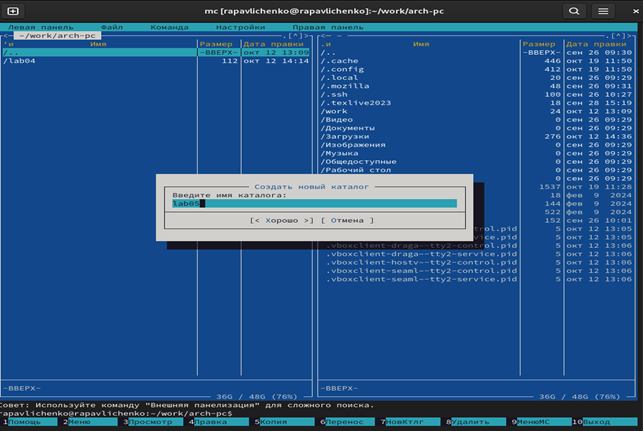
Интерфейс midnight commander

С помощью стрелок и клавиши Enter перейдём в каталог ~/work/arch-pc



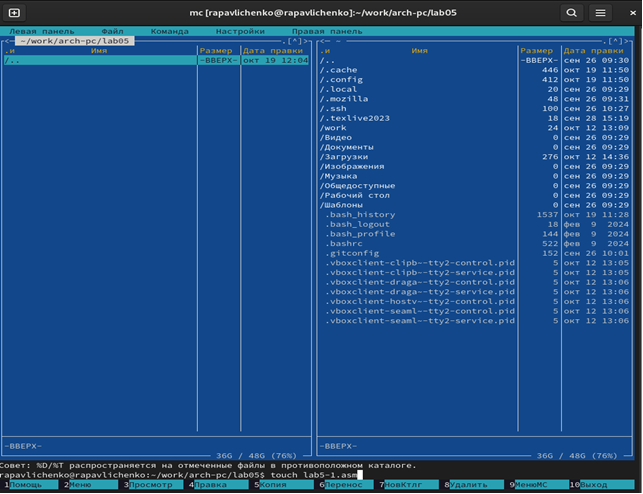
Переход в нужный каталог (~/work/arch-pc)

Создадим папку lab05 с помощью клавиши F7



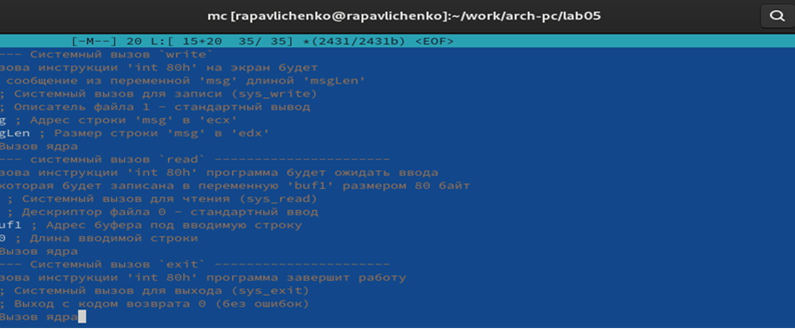
Создание папки

Теперь с помощью команды touch создадим файл lab5-1.asm



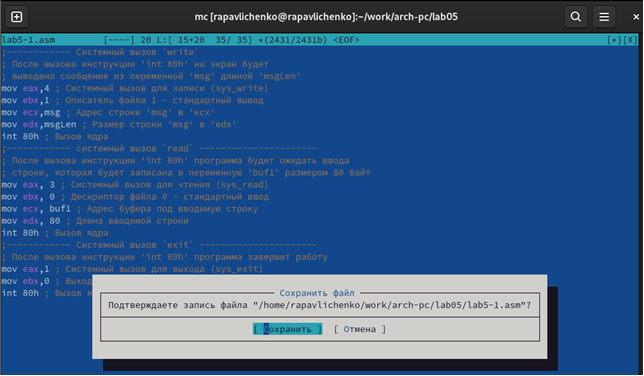
Создание файла lab5-1.asm с помощью команды touch прямо в mc

Теперь с помощью клавиши F4 откроем только что созданный файл и отредактивруем файл, поместим в него следующий код



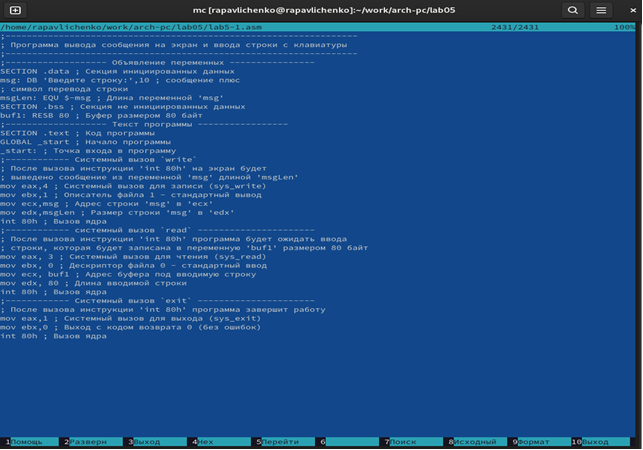
Выбор текстового редактора

Теперь сохраним его клавишей F2



Редактирование файла lab5-1.asm

С помощью функциональной клавиши F3 откроем файл lab5-1.asm для просмотра. Убедимся, что файл содержит текст программы.



Проверка успешного редактирования

Теперь скомпилируем его

Компиляция файла с помощью nasm

Компиляция файла с помощью nasm

И соберём

Сборка исполняемого файла с помощью ld

Сборка исполняемого файла с помощью ld

После этого запустим получившийся исполняемый файл

Запуск исполняемого файла

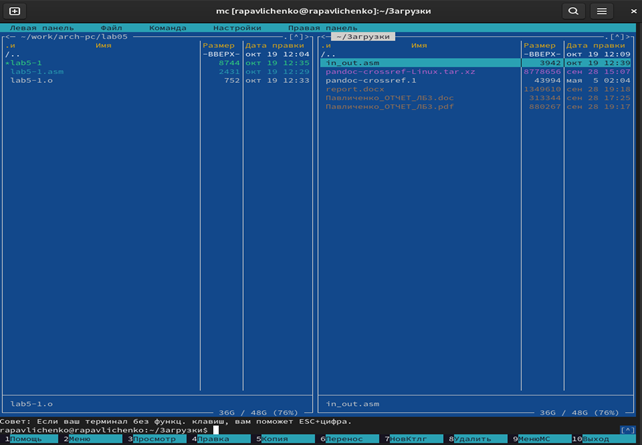
Запуск исполняемого файла

Теперь введём ФИО

Взаимодействие с программой

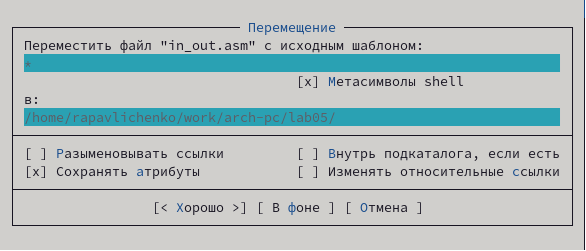
Взаимодействие с программой

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



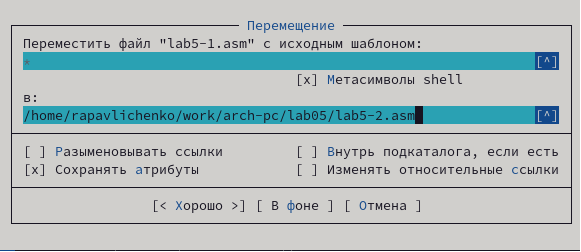
Открытие папки с файлом in\_out.asm в правой панели

Скопируем его в нашу рабочую папку с помощью F6



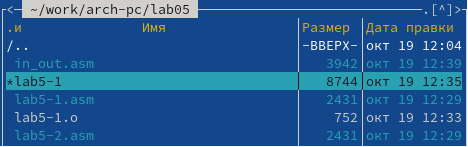
Копирование файла c помощью F6

Теперь сделаем копию файла lab5-1.asm с помощью команды F5. Назовём копию lab5-2.asm



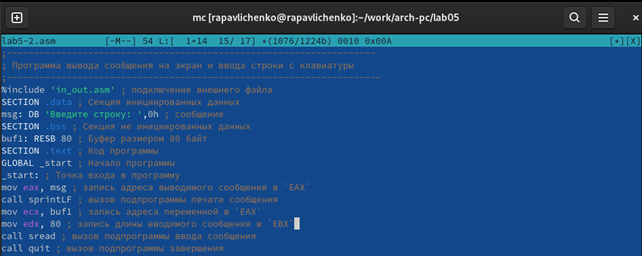
Копирование файла c помощью F5

Теперь наша папка выглядит следующим образом



Текущий вид рабочей папки

Откроем в текстовом редакторе файл lab5-2.asm и напишем туда следующий код



Редактирование файла lab5-2.asm

После чего создадим исполняемый файл с помощью nasm и ld

Создание исполняемого файла

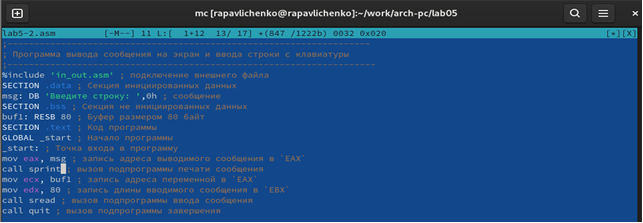
Создание исполняемого файла

Запустим созданный файл и введем ФИО

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



Изменение файла lab5-2.asm

Точно также соберём исполняемый файл и запустим его

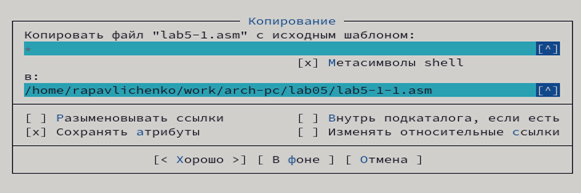
Запуск изменённого файла

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

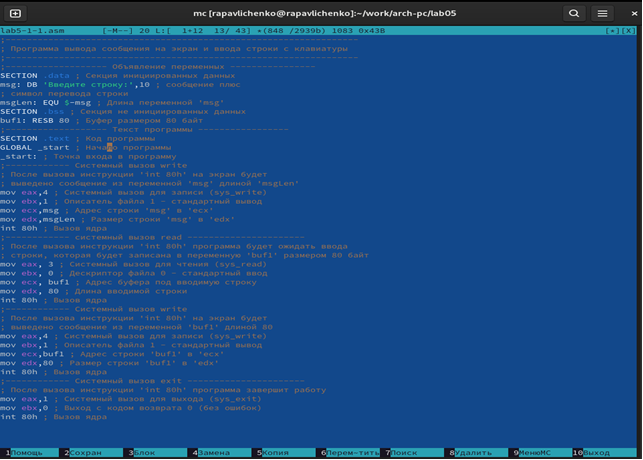
# 3 Выполнение задания для самостоятельной работы

Теперь создадим с помощью F6 копию файла lab5-1.asm



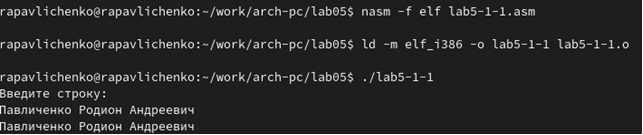
Создание копии файла lab5-1.asm

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):



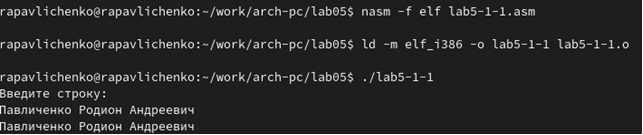
Изменение файла lab5-1-1.asm

Сохраним изменения и создадим исполняемый файл. Запустим его и проверим, что всё работает.



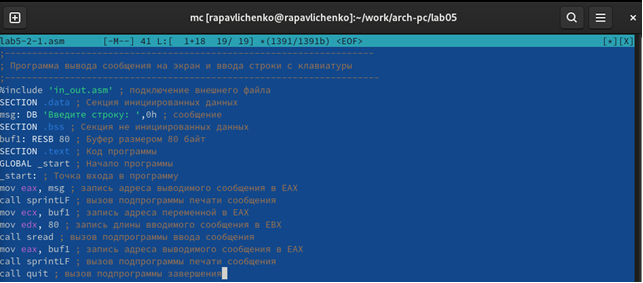
Создание исполняемого файла

Теперь создадим с помощью F5 копию файла lab5-2.asm



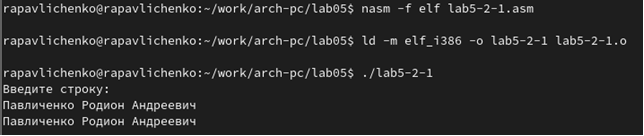
Создание копии файла lab5-2.asm

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. 3.6):



Изменение файла lab5-2-1.asm

Теперь создадим исполняемый файл,запустим программу и убедимся, что она работает



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