Отчёт по лабораторной работе 5

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

Зиборова Вероника Николаевна НММбд-02-24

Содержание

# 1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander, а также освоение инструкций языка ассемблера mov и int.

# 2 Выполнение лабораторной работы

## 2.1 Знакомство с Midnight Commander

Я запустила Midnight Commander и, используя стрелочные клавиши и клавишу Enter, перешла в каталог ~/work/arch-pc. Затем нажала F7 для создания нового каталога lab05.

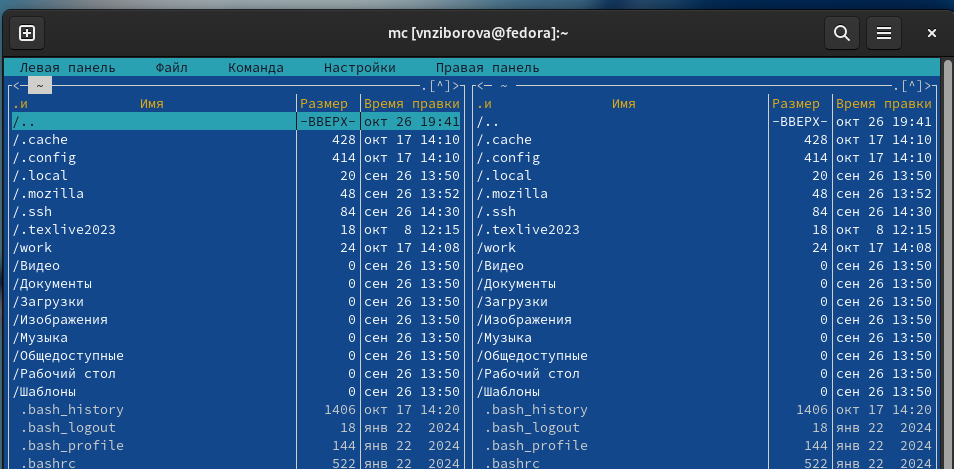


Рис. 1: Запуск Midnight Commander

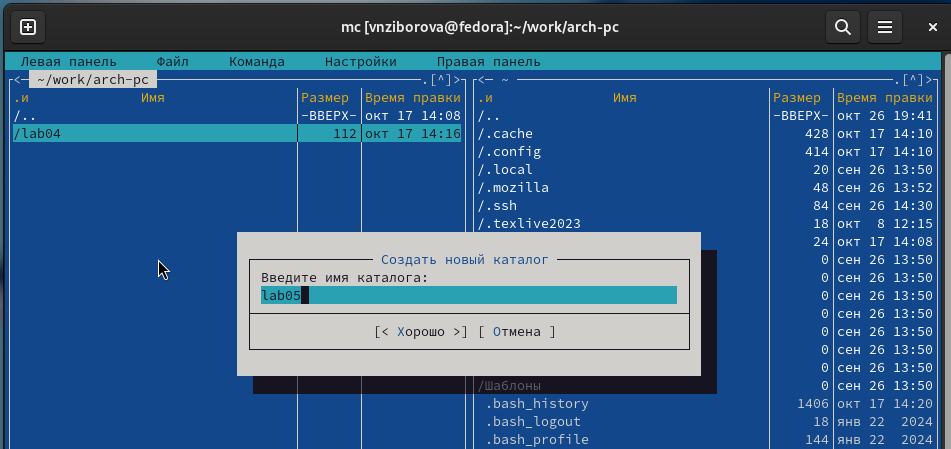


Рис. 2: Создание каталога lab05

С помощью команды touch я создала файл lab05-1.asm.

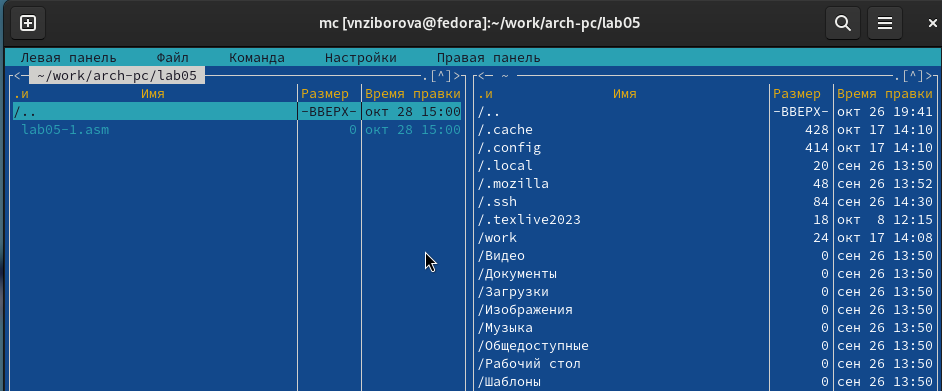


Рис. 3: Создание файла lab05-1.asm

Я открыла файл для редактирования, нажав F4, выбрала редактор mceditor и написала код программы в соответствии с заданием.



Рис. 4: Редактирование программы в файле lab05-1.asm

Для проверки я открыла файл на просмотр, нажав F3, и убедилась, что он содержит правильный код.

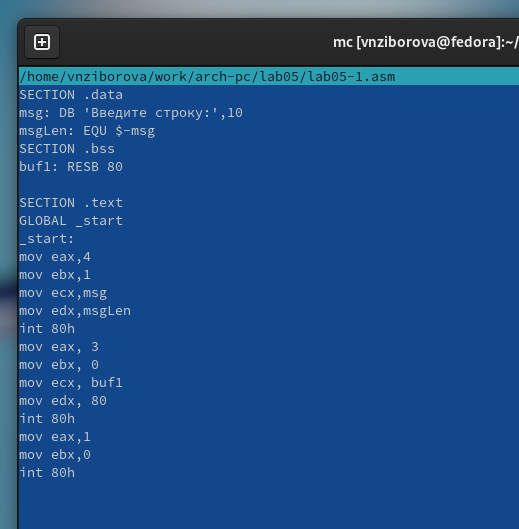


Рис. 5: Просмотр содержимого файла lab05-1.asm

Я скомпилировала программу, сгенерировала объектный файл, произвела компоновку и запустила исполняемый файл, чтобы убедиться в корректности работы.

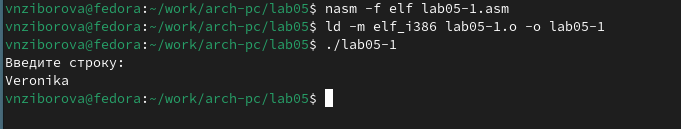


Рис. 6: Запуск программы lab05-1.asm

## 2.2 Подключение внешнего файла in\_out.asm

Я скачала файл in\_out.asm и разместила его в рабочем каталоге. Для копирования я использовала клавишу F5, а для перемещения – клавишу F6.

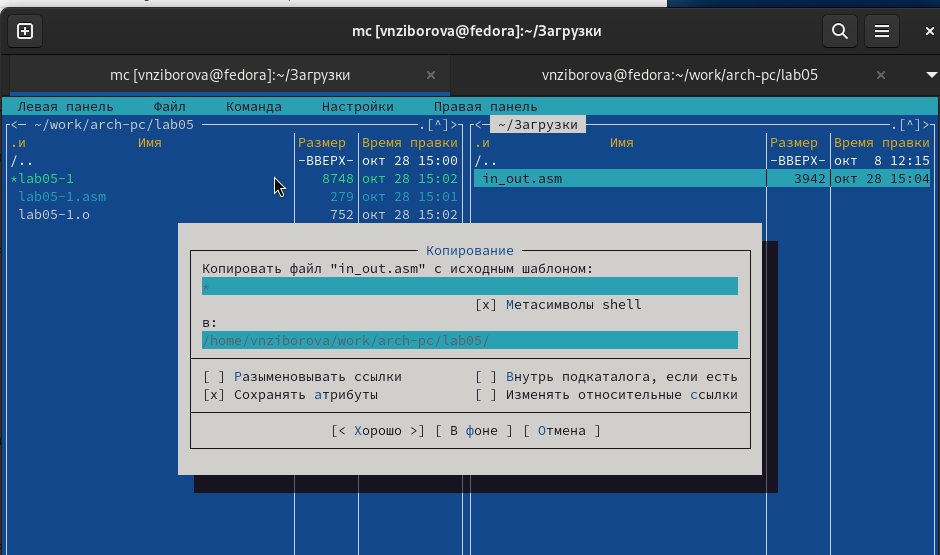


Рис. 7: Копирование файла in\_out.asm в рабочий каталог

Я также скопировала файл lab05-1.asm в новый файл lab05-2.asm.

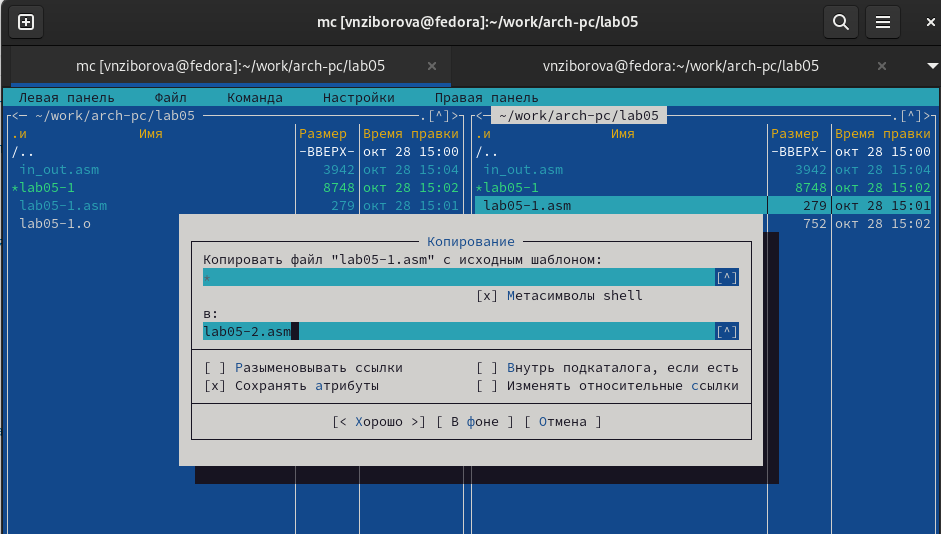


Рис. 8: Копирование файла lab05-1.asm в lab05-2.asm

В файле lab05-2.asm я написала код программы, используя подпрограммы из внешнего файла in\_out.asm. Программа была успешно скомпилирована и протестирована.

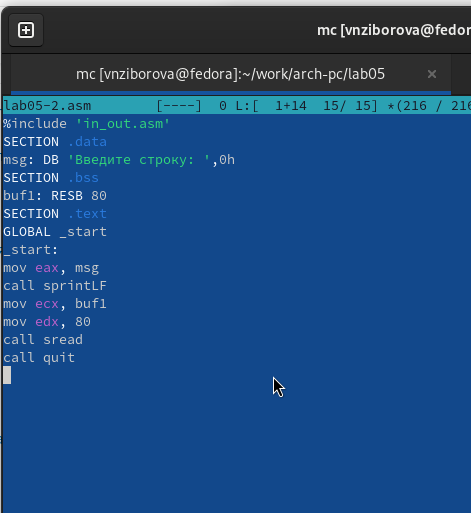


Рис. 9: Программа в файле lab05-2.asm

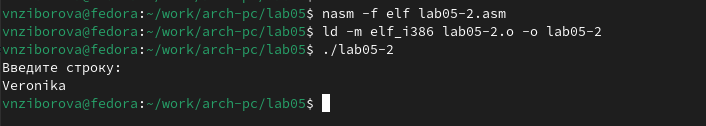


Рис. 10: Запуск программы lab05-2.asm

В файле lab05-2.asm я заменила подпрограмму sprintLF на sprint. После пересборки исполняемого файла, теперь вывод строки не завершается символом новой строки.

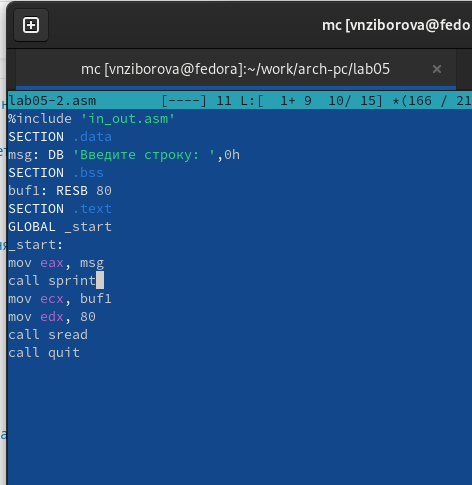


Рис. 11: Обновленная программа в lab05-2.asm

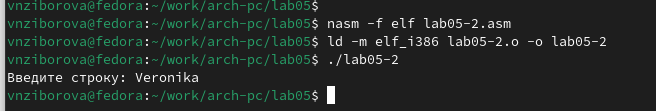


Рис. 12: Запуск обновленной программы lab05-2.asm

## 2.3 Задание для самостоятельной работы

Я скопировала программу lab05-1.asm и изменила код так, чтобы он работал по следующему алгоритму:

* вывести приглашение с текстом “Введите строку:”;
* считать строку с клавиатуры;
* вывести введённую строку на экран.

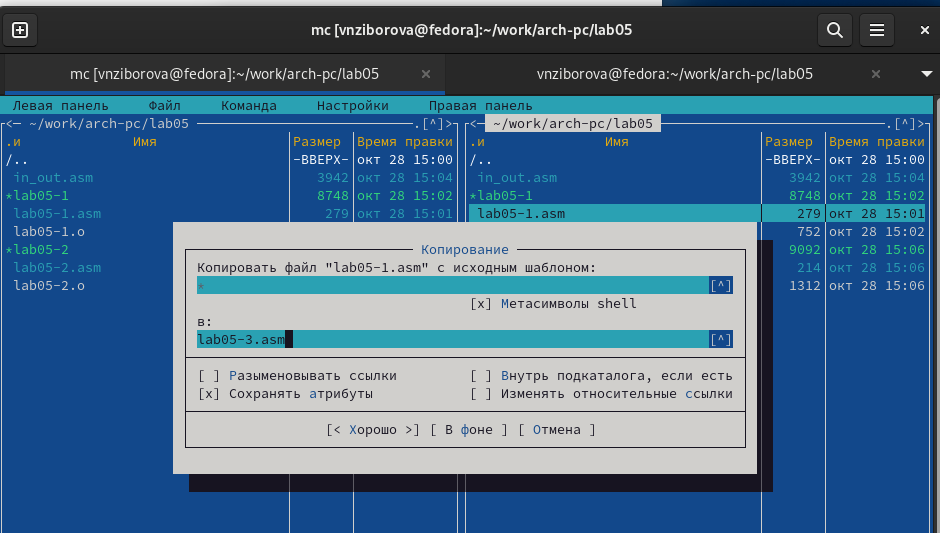


Рис. 13: Копирование файла lab05-1.asm для нового задания

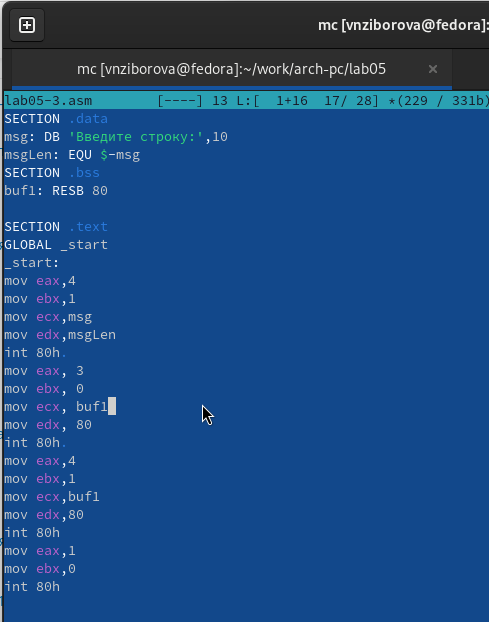


Рис. 14: Код программы в файле lab05-3.asm

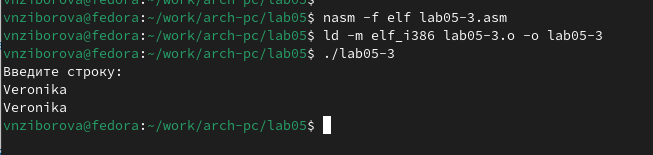


Рис. 15: Запуск программы lab05-3.asm

Аналогично, я скопировала программу lab05-2.asm и внесла изменения в код, теперь используя подпрограммы из файла in\_out.asm.

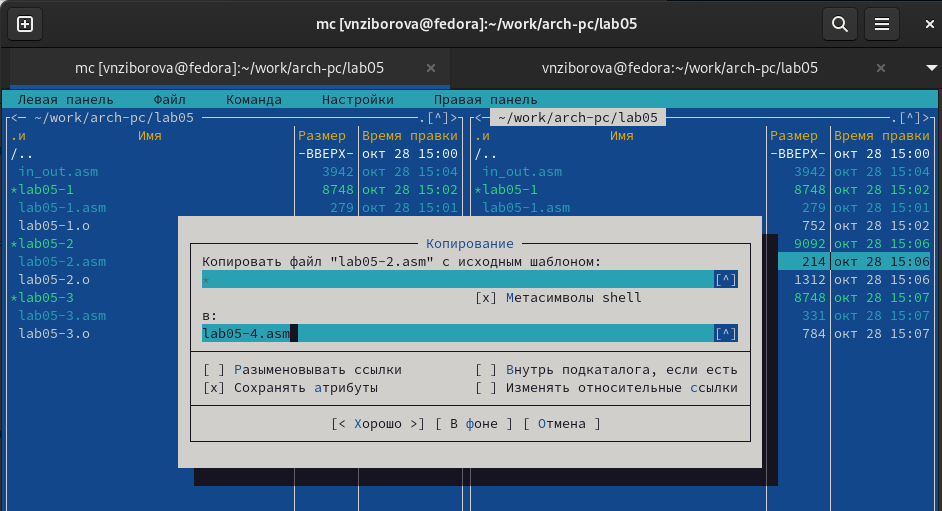


Рис. 16: Копирование файла lab05-2.asm для модификации

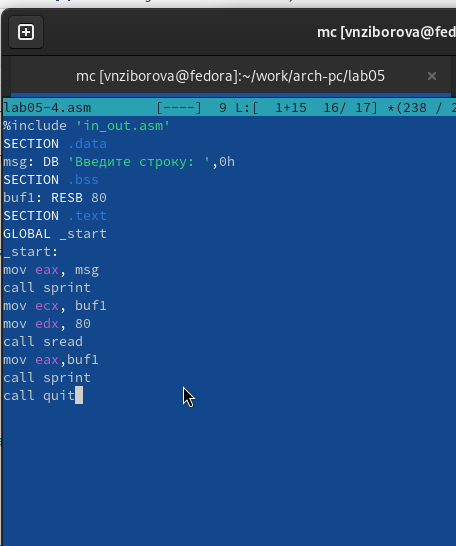


Рис. 17: Код программы в файле lab05-4.asm

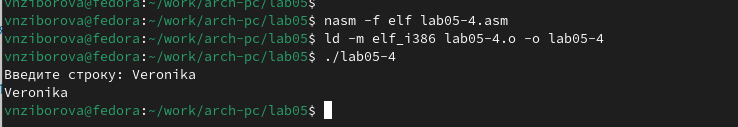


Рис. 18: Запуск программы lab05-4.asm

# 3 Выводы

Я научилась писать базовые ассемблерные программы и освоила ассемблерные инструкции mov и int.

# 4 Вопросы для самопроверки

1. Каково назначение mc?

* Midnight Commander (mc) — это текстовый файловый менеджер, который позволяет пользователям управлять файлами и каталогами в командной строке. Он предоставляет удобный интерфейс для выполнения операций с файлами, таких как копирование, перемещение, удаление и просмотр.

1. Какие операции с файлами можно выполнить как с помощью команд bash, так и с помощью меню (комбинаций клавиш) mc? Приведите несколько примеров.

* В mc можно выполнять операции с файлами как с помощью команд bash, так и с помощью меню. Примеры:
  + **Копирование файла:**
    - Команда bash: cp файл1.txt файл2.txt
    - В mc: выделить файл файл1.txt, нажать F5 и указать имя файл2.txt.
  + **Перемещение файла:**
    - Команда bash: mv файл1.txt папка/
    - В mc: выделить файл файл1.txt, нажать F6 и указать путь папка/.
  + **Удаление файла:**
    - Команда bash: rm файл1.txt
    - В mc: выделить файл файл1.txt, нажать F8.

1. Какова структура программы на языке ассемблера NASM?

* Программа на NASM состоит из трех основных секций:
  + **section .data** — секция для инициализации данных.
  + **section .bss** — секция для объявления неинициализированных данных.
  + **section .text** — секция, содержащая исполняемый код программы.

1. Для описания каких данных используются секции bss и data в языке ассемблера NASM?

* **Секция .data** используется для хранения инициализированных данных, которые имеют фиксированные значения.
* **Секция .bss** используется для объявления неинициализированных данных, которые будут выделены в памяти, но не содержат начальных значений.

1. Для чего используются компоненты db, dw, dd, dq и dt языка ассемблера NASM?

* Эти директивы используются для определения переменных различных типов:
  + **db** (define byte) — определяет байт.
  + **dw** (define word) — определяет слово (2 байта).
  + **dd** (define double word) — определяет двойное слово (4 байта).
  + **dq** (define quad word) — определяет квадратичное слово (8 байт).
  + **dt** (define ten bytes) — определяет десятибайтовое значение.

1. Какое произойдёт действие при выполнении инструкции mov eax, esi?
   * Инструкция mov eax, esi копирует значение из регистра esi в регистр eax. Это означает, что после выполнения этой инструкции регистр eax будет содержать то же значение, что и esi.
2. Для чего используется инструкция int 80h?
   * Инструкция int 80h используется для вызова системных функций в Linux. Она передает управление ядру операционной системы для выполнения системного вызова, например, для работы с файлами, сетевыми операциями и управления процессами.