



Лабораторная работа #9

Основы файлового ввода-вывода в Linux. ИСПОЛЬЗОВАНИЕ СИСТЕМНЫХ ВЫЗОВОВ

[fd, inode, creat, open, write, read, perror, errno, link, unlink, stat, mkdeir, rmdir, chdir, opendir, closedir, readdir]



ЛАБОРАТОРНАЯ РАБОТА #9

Основы файлового ввода-вывода в Linux.

Использование системных вызовов

Цель работы

Повторить фундаментальные основы и концепции файлового ввода-вывода в Linux, а также изучить системные вызовы (низкоуровневые функции) для взаимодействия с файловой системой и научиться использовать низкоуровневый ввод-вывод при написании системных программ.

Требования

- 1) Разработать многофайловый консольный проект на C/C++ согласно варианту задания с использованием шаблона (паттерна) проектирования MVC.
- 2) Предусмотреть ещё один способ инициализации данных и вывода результата работы программы – с помощью файлового ввода-вывода. Программа должна дополнительно выводить на консоль исходные данные и конечный результат.
- 3) ЗАПРЕЩАЕТСЯ в программе использовать под любым предлогом ГЛОБАЛЬНЫЕ переменные!
- 4) Для повышения производительности программы и закрепления навыков работы с памятью везде, где это возможно, необходимо использовать ДИНАМИЧЕСКОЕ выделение и освобождение памяти, а также осуществлять работу через УКАЗАТЕЛИ.
- 5) Каждое задание оформить в виде отдельной бизнес-функции. Все функции должны быть сгруппированы по соответствующим отдельным файлам и вынесены в отдельную библиотеку.

- 6) Все функции должны быть самодостаточные, т.е. при их разработке необходимо придерживаться принципа ***Single Responsibility Principle***.
- 7) При выполнении задания разрешается использовать *IDE*, а также задействовать любой текстовый редактор (к примеру, ***gedit***) и набор компиляторов GNU Compiler Collection (GCC), в частности, компиляторы языков программирования C/C++ ***gcc/g++***, а также утилиту для создания файлов-архивов ***ar***.
- 8) Для автоматизации сборки проекта необходимо использовать утилиту-автосборщик – ***GNU make***.
- 9) При разработке программ придерживайтесь соглашений по написанию кода на C/C++ (Code-Convention).
- 10) Контрольные вопросы по лабораторной работе и ответы на них должны быть записаны в конспект.

Основное задание

Произвести рефакторинг основного задания предыдущей лабораторной работы: пользовательскую библиотеку функций ввода-вывода переписать с использованием низкоуровневых функций, которые представляет само ядро ОС.

Индивидуальное задание

С использованием низкоуровневого ввода-вывода (т.е. системных вызовов) разработать модульное консольное приложение согласно варианту задания.

- 1) Создать и заполнить файл случайными целыми числами. Отсортировать содержимое файла по возрастанию.
- 2) Прочитать текст C/C++ программы и записать в другой файл в обратном порядке символы каждой строки.
- 3) Прочитать текст C/C++ программы и в каждом слове длиннее двух символов все строчные символы заменить прописными.
- 4) В файле, содержащем фамилии студентов и их оценки, записать прописными буквами фамилии тех студентов, которые имеют средний балл более "7".
- 5) Файл содержит символы, слова, целые числа и числа с плавающей запятой. Определить все данные, тип которых вводится из командной строки.
- 6) Из файла удалить все слова, содержащие от трех до пяти символов, но при этом из каждой строки должно быть удалено только максимальное четное количество таких слов.

- 7) Прочитать текст C/C++ программы и удалить из него все “лишние” пробелы и табуляции, оставив только необходимые для разделения операторов.
- 8) Из текста C/C++ программы удалить все виды комментариев.

Дополнительное задание

Разработать программу-тест, которая для своей работы использует только системные вызовы и предоставляет следующие возможности:

- создание любого файла для записи и чтения; уничтожение указанного файла;
- создание (уничтожение) и чтение любого каталога;
- получение данных о файлах;
- запись (чтение) в файл информации, которая включает в себя все доступные примитивные типы данных языка C/C++, а также текстовых данных (не менее трёх строк).

Best of LUCK with it, and remember to HAVE FUN while you're learning :)

 Victor Ivanchenko



Контрольные вопросы

1. Основные концепции файлового ввода-вывода с точки зрения ОС Linux (таблица файлов, дескрипторы файлов (*fd*), обычные файлы, позиция в файле, смещение в файле, длина файла, усечение, информационный узел, номер *inode*)?
2. Абстракция файла и файлового ввода-вывода на уровне ядра ОС Linux (*nix)?
3. Каталоги и ссылки в Linux (каталог, ссылка, корневой каталог, полный путь, относительный путь, записи каталога, разрешение полного пути, кэш *dentry*, абсолютный полный путь, рабочий каталог, жёсткие ссылки, символические ссылки, специальные файлы)?
4. Системные вызовы: ограничения системных вызовов, коды возврата системных вызовов, режим ядра, режим пользователя?
5. Использование системных вызовов для низкоуровневой работы с каталогами и файлами: *mkdir*, *rmdir*, *chdir*, *opendir*, *closedir*, *readdir*, *readlink*, *link*, *unlink*, *stat*, ... ?
6. Использование системных вызовов для низкоуровневой работы с файлом: *open*, *creat*, *write*, *read*, *close*, ... ?
7. Функции библиотеки C для получения описания возникших ошибок файлового ввода-вывода: *perror*, *strerror*, *sys_errlist*, переменная *errno*?