

Программа лабораторной работы по курсу ОС

по теме:

Файловые системы Linux-подобных ОС

Проанализировать функциональное назначение структурных элементов дерева ФС.
Определить размещение корневого каталога (корневой ФС).

1. Ознакомиться с типами файлов исследуемой ФС. Применяя утилиту *ls*, отфильтровать **по одному примеру** каждого типа файла используемой вами ФС. Комбинируя различные ключи утилиты рекурсивно просканировать все дерево, анализируя крайнюю левую позицию выходной информации, полученной посредством *ls -l*. Результат записать в выходной файл **с указанием полного пути** каждого примера. Выполнить задание сначала в консоли построчно, выбирая необходимые сочетания ключей (в командной строке), а затем оформить как скрипт с задаваемым в командной строке именем выходного файла

Файлы системы могут быть следующих типов:

- Обычный файл
- b Специальный файл блочного устройства.
- c Файл символического устройства
- d Директория
- l Символьная ссылка
- p FIFO
- s Сокет

2. Получить все *жесткие ссылки* на заданный файл, находящиеся в разных каталогах пользовательского пространства (разными способами, не применяя утилиты *file* и *find*). Использовать конвейеризацию и фильтрацию. Оформить в виде скрипта.

3. Проанализировать все возможные способы формирования *символьных ссылок* (*ln*, *link*, *sr* и т.д.), продемонстрировать их экспериментально. Предложить скрипт, подсчитывающий и перечисляющий все полноименные символьные ссылки на файл, размещаемые в разных местах файлового дерева.

4. Получить все символьные ссылки на заданный в качестве входного параметра файл, не используя *file* (разными способами, не применяя утилиту *file*).

5. Изучить утилиту *find*, используя ее ключи получить расширенную информацию о всех типах файлов. Создать примеры вложенных команд.

6. Проанализировать *содержимое заголовка файла* (для разных типов файла), а также файла-каталога с помощью утилит *od* (POSIX, Linux) и *hexdump* (Linux). Если доступ к файлу-каталогу возможен (для отдельных модификаций POSIX-совместимых ОС), проанализировать изменение его содержимого при различных операциях над элементами, входящими в его состав (файлами и подкаталогами).

7. Определить размер вновь созданного каталога. Где задается ограничение? Как меняется размер каталога в зависимости от размещения в нем новых различных объектов, а также их дальнейшего удаления. Покажите это экспериментально. Создайте скрипт для упрощения экспериментов.

8. Ознакомиться с содержимым */etc/passwd*, */etc/shadow*, с утилитой */usr/bin/passwd*, проанализировать права доступа к этим файлам.
9. Исследовать права владения и доступа, а также их сочетаемость
 - 9.1. Привести примеры применения утилит *chmod*, *chown* к специально созданному для этих целей отдельному каталогу с файлами.
 - 9.2. Расширить права исполнения экспериментального файла с помощью флага **SUID**.
 - 9.3. Экспериментально установить, как формируются итоговые права на использование файла, если права пользователя и группы, в которую он входит, различны.
 - 9.4. Сопоставить возможности исполнения наиболее часто используемых операций, варьируя правами доступа к файлу и каталогу.
10. Разработать «программу-шлюз» для доступа к файлу другого пользователя при отсутствии прав на чтение информации из этого файла. Провести эксперименты для случаев, когда пользователи принадлежат одной и разным группам. Сравнить результаты. Для выполнения задания применить подход, аналогичный для обеспечения функционирования утилиты */usr/bin/passwd* (манипуляции с правами доступа, флагом **SUID**, а также размещением файлов).
11. Применяя утилиту *df* и аналогичные ей по функциональности утилиты, а также информационные файлы типа *fstab*, получить информацию о файловых системах, *возможных* для монтирования, а также *установленных* на компьютере *реально*.
 - 11.1. Привести информацию об исследованных *утилитах* и *информационных файлах* с анализом их содержимого и *форматов*.
 - 11.2. Привести образ диска с точки зрения состава и размещения всех ФС на испытуемом компьютере, а также образ полного дерева ФС, **включая присоединенные ФС съемных и несъемных носителей**. Проанализировать и указать формат таблицы монтирования.
 - 11.3. Привести «максимально возможный путь» (дерево) ФС, проанализировать, где указывается ограничение
12. Проанализировать и пояснить принцип работы утилиты *file*.
 - 12.1. Привести алгоритм её функционирования на основе информационной базы, размещение и полное имя которой указывается в описании утилиты в технической документации ОС (как правило, */usr/share/file/magic.**), а также содержимого заголовка файла, к которому применяется утилита. Определить, где находятся магические числа и иные характеристики, идентифицирующие тип файла, применительно к *исполняемым* файлам, а также файлам других типов.
 - 12.2. Утилиту *file* выполнить с разными ключами.
 - 12.3. Привести экспериментальную попытку с добавлением в базу собственного типа файла и его дальнейшей идентификацией. Описать эксперимент и привести последовательность действий для расширения функциональности утилиты *file* и

возможности встраивания дополнительного типа файла в ФС (согласовать содержимое информационной базы и заголовка файла нового типа