

Лабораторная работа № 3

ФАЙЛОВАЯ СИСТЕМА LINUX

1. Цель работы

Целью работы является изучение структуры файловой системы ОС LINUX, изучение команд создания, удаления, модификации файлов и каталогов, функций манипулирования данными.

2. Задачи работы

– Закрепление, углубление и расширение знаний студентов при использовании файловой системы ОС Linux.

– Приобретение умений и навыков работы с функциями файловой системы по модификации файлов и каталогов в операционной системе Linux.

– Выработка способности логического мышления, осмысления полученных результатов при применении файловых команд ОС Linux.

3. Теоретическая часть

Файловая структура системы LINUX

Файловая система Linux, в отличие от операционных систем семейства Windows не разделена по томам (дискам, устройствам), а имеет единую древовидную структуру, в основе которой лежит корневой каталог. Корневой каталог – это уровень файловой системы, выше которого по дереву каталогов подняться невозможно. В Linux корневой каталог обозначается как / (именно / – слэш, а не \ – обратный слэш). Система позволяет устанавливать много корневых каталогов. Так например для некоторого пользователя ftp /home будет корневым каталогом и при обращении к клиенту ftp на смену каталога на корневой пользователь будет попадать в /home.

Возникает вопрос, а как тогда разные физические устройства участвуют в формировании единой файловой системы? Сделаем небольшой экскурс в историю. В то время, когда создавалась ОС Юникс устройства – накопители информации представляли собой ящик размером с письменный стол и назывались магнитными барабанами. В то время не было необходимости подключать и отключать его по нескольку раз в час. Поэтому не был выработан и механизм быстрой смены. Для подключения любого устройства к файловой системе используется так называемая точка монтирования – каталог, все вложенные уровни которого являются файловой системой на устройстве-носителе. Например, при монтировании дискеты обычно используется каталог /media/floppy. То есть, все каталоги и файлы, находящиеся внутри /media/floppy на самом деле содержатся на дискете, вставленной в дисковод компьютера. Для подключения, или монтирования, устройств используется специальная команда mount. Таким образом подключаются и сетевые файловые системы, то есть такие системы, которые реально находятся где-то на сервере сети, однако различий в работе с ними пользователь не ощущает и видит сетевые файлы и каталоги, как если бы они были расположены на локальном компьютере.

Есть у файловой системы Linux и еще одна особенность. Каждому пользователю в ней выделяется домашний каталог – специальный каталог, необходимый для хранения пользователем своих личных данных. При входе пользователя в систему, он сразу оказывается в своем домашнем каталоге. Обычно права доступа к домашнему каталогу пользователя выставлены таким образом, что доступ к каталогу запрещен всем кроме владельца и администратора.

В операционной системе LINUX файлами считаются обычные файлы, каталоги, а также специальные файлы, соответствующие периферийным устройствам (каждое устройство представляется в виде файла). Доступ ко всем файлам однотипный, в том числе, и к файлам периферийных устройств. Такой подход обеспечивает независимость программы пользователя от особенностей ввода/вывода на конкретное внешнее устройство.

Файловая структура LINUX имеет иерархическую древовидную структуру, показанную в таблице:

Имя каталога	Описание
/bin	в этом каталоге находятся основные исполняемые файлы, жизненно необходимые для функционирования системы
/boot	содержит ядро операционной системы и карты загрузки, а также конфигурационные файлы загрузчиков (lilo, grub)
/dev	содержит файлы, которые являются интерфейсом с периферийными устройствами
/etc	содержит основные файлы настроек приложений Linux
/home	содержит домашние папки пользователей
/lib	содержит основные библиотеки, необходимые для нормальной работы системы
/lost+found	информация, восстановленная при проверке файловой системы на наличие ошибок
/media	точки монтирования отключаемых устройств (usb-диски, CD, floppy)
/mnt	точки монтирования ISO-образов, сетевых файловых систем, других постоянных файловых систем
/opt	альтернатива usr, для коммерческого ПО или ПО, не входящего в основной дистрибутив
/proc	внутри этого каталога находится виртуальная файловая система <i>proc</i> , создаваемая ядром Linux “на лету”. Содержит общую информацию о системе и подробную о процессах.
/root	домашний каталог пользователя root
/sbin	утилиты суперпользователя
/srv	файлы, выкладываемые для доступа всевозможных внешних служб (например, tftp)
/sys	внутри этого каталога также находится виртуальная файловая система, только она содержит подробную информацию о процессах
/tmp	в этом каталоге находятся временные файлы, используемые запущенными в данный момент процессами
/usr	программы, библиотеки и другие данные пользовательских приложений
/var/log	содержит файлы журналов

Текущий каталог – это каталог, в котором в данный момент находится пользователь. При наличии прав доступа, пользователь может перейти после входа в систему в другой каталог. Текущий каталог обозначается точкой (.); родительский каталог, которому принадлежит текущий, обозначается двумя точками (..).

Полное имя файла может включать имена каталогов, включая корневой, разделенных косой чертой, например: /home/student/file.txt. Первая косая черта обозначает корневой каталог, и поиск файла будет начинаться с него, а затем в каталоге home, затем в каталоге student.

Один файл можно сделать принадлежащим нескольким каталогам. Для этого используется команда **ln (link)**, создающая «жесткую ссылку»:

ln <имя файла 1> <имя файла 2>

Имя 1-го файла – это полное составное имя файла, с которым устанавливается связь; имя 2-го файла - это полное имя файла в новом каталоге, где будет использоваться эта связь. Новое имя может не отличаться от старого. Каждый файл может иметь несколько связей, т.е. он может использоваться в разных каталогах под разными именами.

Жесткие ссылки можно создавать только для файлов, но не для директорий. Исключение составляют специальные записи каталогов, указывающие на саму директорию и на ее родительскую директорию (. и ..) и представляющие собой жесткие ссылки, необходимые для подсчета числа поддиректорий. Жесткие ссылки можно использовать только в пределах одной файловой системы, поскольку они являются указателями на дескрипторы inode, которые уникальны только в пределах отдельной файловой системы. Если файл имеет несколько жестких ссылок, то он удаляется только тогда, когда удаляется последняя ссылка, указывающая на его inode, и счетчик ссылок сбрасывается до 0.

Команда **ln** с аргументом -s создает символическую связь («мягкую ссылку»):

ln -s <имя файла 1> <имя файла 2>

Здесь имя 2-го файла является именем символической связи.

«Мягкая ссылка» (symlink) просто указывает на имя другого файла или директории, а не на его inode. В отличие от жестких ссылок, мягкие ссылки могут указывать на объекты разных файловых систем. Удаление мягкой ссылки не приводит к удалению файла или директории, на которую она указывает, а удаление целевого объекта не приводит к автоматическому удалению мягких ссылок

В LINUX различаются 3 уровня доступа к файлам и каталогам:

- 1) доступ владельца файла;
- 2) доступ группы пользователей, к которой принадлежит владелец файла;
- 3) остальные пользователи.

Для каждого уровня существуют свои байты атрибутов, значение которых расшифровывается следующим образом:

- r – разрешение на чтение;
- w – разрешение на запись;
- x – разрешение на выполнение;
- – отсутствие разрешения.

Первый символ байта атрибутов определяет тип файла и может интерпретироваться со следующими значениями:

- – обычный файл;
- d – каталог;
- l – символическая связь;

b – блок-ориентированный специальный файл, который соответствует таким периферийным устройствам, как накопители на магнитных дисках;

c – байт-ориентированный специальный файл, который может соответствовать таким периферийным устройствам как принтер, терминал.

В домашнем каталоге пользователь имеет полный доступ к файлам (READ, WRITE, EXECUTE; r, w, x).

Атрибуты файла можно просмотреть командой **ls -l** и они представляются в следующем формате:

d	rwX	rwX	rwX	
				Доступ для остальных пользователей
				Доступ к файлу для членов группы
				Доступ к файлу владельца
				Тип файла (директория)

Пример. Командой **ls -l** получим листинг содержимого текущей директории student:

```
- rwX --- --- 2 student 100 Mar 10 10:30 file_1
- rwX --- r-- 1 adm    200 May 20 11:15 file_2
- rwX --- r-- 1 student 100 May 20 12:50 file_3
```

После байтов атрибутов на экран выводится следующая информация о файле:

- число связей файла;
- имя владельца файла;

- размер файла в байтах;
- дата создания файла (или модификации);
- время;
- имя файла.

Атрибуты файла и доступ к нему, можно изменить командой:

chmod <коды защиты> <имя файла>

Коды защиты могут быть заданы в числовом или символьном виде. Для символьного кода используются:

- знак плюс (+) - добавить права доступа;
- знак минус (-) - отменить права доступа;
- r,w,x - доступ на чтение, запись, выполнение;
- u,g,o - владельца, группы, остальных.

Коды защиты в числовом виде могут быть заданы в восьмеричной форме. Для контроля установленного доступа к своему файлу после каждого изменения кода защиты нужно проверять свои действия с помощью команды **ls -l**.

Примеры:

chmod g+rw,o+r file.1 - установка атрибутов чтения и записи для группы и чтения для всех остальных пользователей;

ls -l file.1 - чтение атрибутов файла;

chmod o-w file.1 - отмена атрибута записи у остальных пользователей;

>letter - создание файла letter. Символ > используется как для переадресации, так и для создания файла;

cat - вывод содержимого файла;

cat file.1 file.2 > file.12 - конкатенация файлов (объединение);

mv file.1 file.2 - переименование файла file.1 в file.2;

mv file.1 file.2 file.3 directory - перемещение файлов file.1, file.2, file.3 в указанную директорию;

rm file.1 file.2 file.3 - удаление файлов file.1, file.2, file.3;

cp file.1 file.2 - копирование файла с переименованием;

mkdir namedir - создание каталога;

rm dir_1 dir_2 - удаление каталогов dir_1 dir_2;

ls [acdfgilqrstv CFR] namedir - вывод содержимого каталога; если в качестве namedir указано имя файла, то выдается вся информация об этом файле. Значения аргументов:

- l — список включает всю информацию о файлах;
- t — сортировка по времени модификации файлов;
- a — в список включаются все файлы, в том числе и те, которые начинаются с точки;
- s — размеры файлов указываются в блоках;
- d — вывести имя самого каталога, но не содержимое;
- r — сортировка строк вывода;

- i – указать идентификационный номер каждого файла;
- v – сортировка файлов по времени последнего доступа;
- q – непечатаемые символы заменить на знак ?;
- c – использовать время создания файла при сортировке;
- g – то же что -l, но с указанием имени группы пользователей;
- f – вывод содержимого всех указанных каталогов, отменяет флаги -l, -t, -s, -r и активизирует флаг -a;
- C – вывод элементов каталога в несколько столбцов;
- F – добавление к имени каталога символа / и символа * к имени файла, для которых разрешено выполнение;
- R – рекурсивный вывод содержимого подкаталогов заданного каталога.

cd <namedir> - переход в другой каталог. Если параметры не указаны, то происходит переход в домашний каталог пользователя.

pwd - вывод имени текущего каталога;

grep [-vcilns] [шаблон поиска] <имя файла> - поиск файлов с указанием или без указания контекста (шаблона поиска).

Значение ключей:

- v – выводятся строки, не содержащие шаблон поиска;
- c – выводится только число строк, содержащих или не содержащих шаблон;
- i – при поиске не различаются прописные и строчные буквы;
- l – выводятся только имена файлов, содержащие указанный шаблон;
- n – перенумеровать выводимые строки;
- s – формируется только код завершения.

Примеры.

1. Напечатать имена всех файлов текущего каталога, содержащих последовательность "student" и имеющих расширение .txt:

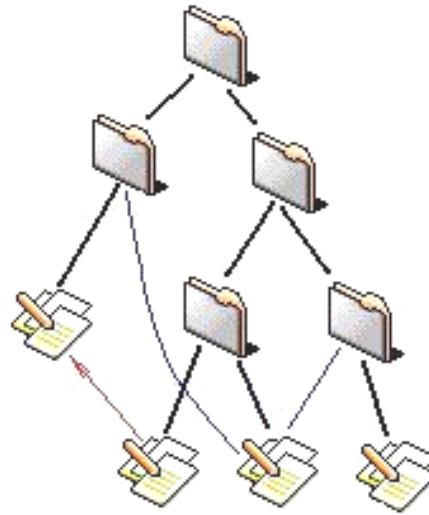
grep -l student *.txt

2. Определить имя пользователя, входящего в ОС LINUX с терминала tty23:



who | grep tty23

4. Задание на лабораторную работу

1. Ознакомиться с файловой структурой ОС LINUX. Изучить команды работы с файлами.
2. Используя команды ОС LINUX, создать два текстовых файла.
3. Полученные файлы объединить в один файл и его содержимое просмотреть на экране.
4. Создать структуру каталогов, соответствующую рисунку ниже.

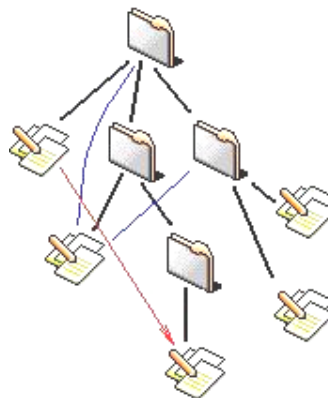


На рисунке представлены:

каталоги	
файлы	

Черными линиями представлена вложенность файлов/подкаталогов в каталоги. Синими линиями представлены ссылки. Красными линиями - символические ссылки. Стрелка на красной линии указывает на целевой файл ссылки.

5. Создать следующую структуру каталогов:



6. Вывести полную информацию обо всех файлах в выбранном каталоге и проанализировать уровни доступа.

7. Добавить для трех файлов право выполнения членам группы и остальным пользователям.

7. Просмотреть атрибуты файлов.

8. Произвести поиск заданной последовательности символов в файлах текущей директории и получить перечень соответствующих файлов.

9. Произвести монтирование флешки в выбранную директорию.
10. Просмотреть список файлов на флешке.
11. Размонтировать флешку.

Контрольные вопросы

1. Что считается файлами в ОС LINUX?
2. Объясните назначение связей с файлами и способы их создания.
3. Что определяет атрибуты файлов и каким образом их можно просмотреть и изменить?
4. Какие методы создания и удаления файлов, каталогов Вы знаете?
5. В чем заключается поиск по шаблону?
6. Какой командой можно получить список работающих пользователей и сохранить его в файле?