

Основы системного администрирования и сетевых технологий

УРОК №3

1. Файлы и каталоги

С точки зрения пользователя, файловая система - это место, где хранятся файлы, а также способ их организации и доступа к ним. Классическая файловая система имеет иерархическую структуру, в которой каждый файл определится уникальным путем к нему. Большинство файловых систем используют каталоги в качестве основного организационного принципа. Каталог - это список ссылок на файлы или другие каталоги. Хотя принято говорить, что каталог содержит файлы или другие каталоги, на самом деле он просто ссылается на них. Если в каталоге есть ссылка на другой каталог, то он называется подкаталогом или вложенным каталогом. Один и тот же файл может иметь ссылки в нескольких каталогах одновременно, что делает доступ к файлу более удобным.

2. Допустимые имена файлов и каталогов

Имена файлов и каталогов на компьютере не могут быть длиннее 256 символов и не могут содержать символ «/», так как этот символ используется для разделения каталогов. Важно помнить, что в именах файлов и каталогов прописные и строчные буквы различаются, то есть «file.txt» и «File.txt» - это два разных файла.

Также существуют некоторые символы, которые можно использовать в именах файлов и каталогов, но их использование не рекомендуется, так как они могут вызвать проблемы при работе с файлами и программами.

Это специальные символы: «*», «&», «>», «<», «;», «(», «)», символы пробела и табуляции. Эти символы имеют особое значение для любой командной оболочки. Избегайте использования русских символов в именах файлов. В Linux нет ограничений на количество точек в имени файла и расширения файла не являются обязательными

3. Дерево каталогов

В системе файлов Linux есть только один корневой каталог, который имеет название «/». Местоположение каждого каталога в дереве каталогов определяется полным путем. Полный путь к файлу или каталогу в Linux начинается от корневого каталога и содержит все промежуточные каталоги, которые нужно пройти, чтобы добраться до искомого места. Важно отметить, что полный путь всегда начинается с символа «/». Каждая программа в Linux работает в своем текущем каталоге файловой системы.

Это означает, что все команды, которые вы вводите в командной строке, наследуют текущий каталог программы и выполняются в этом каталоге. Относительный путь - это путь к файловой системе, который не начинается с корневого каталога. В отличие от полного пути, относительный путь начинается от текущего каталога. Если имя объекта начинается с символа «/», то это полный путь, а в любом другом случае - относительный.

В Linux каждый пользователь обязательно имеет свой каталог, который становится текущим после регистрации - это домашний каталог. Обычно имя домашнего каталога совпадает с именем пользователя и находится в каталоге /home.

4. Шаблоны

Часто возникает необходимость выполнять операции сразу над несколькими файлами, например, копирование. Для того чтобы указать имена нескольких файлов, не перечисляя их вручную, используются шаблоны. Шаблон - это строка текстового формата, который используется в процедурах текстового поиска. Символы в шаблоне разделяются на обычные и специальные. Обычные символы соответствуют таким же символам в строке, а специальные символы обрабатываются следующим образом:

- Если шаблон состоит только из обычных символов, то он соответствует только одной строке, которая состоит из тех же символов в том же порядке.
- Если шаблон состоит из символа «*», то он соответствует любой строке любой длины, включая пустую строку.
- Если шаблон состоит из символа «?», то он соответствует любой строке длиной в один символ.
- Если шаблон заключен в квадратные скобки, то он соответствует строке длиной в один символ, причем этот символ должен встречаться среди заключенных в скобки символов. Например, шаблону [baг] соответствуют только строки «b», «a» и «г». Символы между квадратными скобками можно задавать в виде диапазона, используя символ «-» между начальным и конечным символом.
- Если после символа «[» следует «!», то шаблон соответствует строке из одного символа, который не перечислен между скобками.
- Если шаблон состоит из нескольких частей, то он соответствует строке, которую можно разбить на столько же подстрок,

4. Шаблоны

сколько частей в шаблоне. Первая подстрока соответствует первой части шаблона, вторая - второй и т.д. Например, шаблону `a*b?c` соответствует любая строка, которая начинается с буквы «а», затем может содержать любое количество других символов, за которыми следует буква «b» и через один символ - буква «с». Например, `aasgfbzс`. Тема: работа с файлами и шаблонами в операционных системах.

5. Команды для работы с файлами и каталогами

Команда `pwd`

Чтобы узнать, где находится пользователь в дереве каталогов, т.е. название текущей папки, нужно ввести команду «`pwd`» (`print working directory` - вывести рабочий каталог) – печать имени текущего каталога. Это позволит вывести название текущей папки на экран.

Команда `cd`

В каждом созданном каталоге могут быть файлы или другие каталоги. Чтобы не вводить длинные имена файлов, можно сделать текущий каталог с помощью команды `cd` [полное_имя_каталога]. Команда `cd ~` возвращает домашний каталог пользователя в качестве текущего каталога из любого каталога. Команда `cd` без параметров эквивалентна `cd ~`. Команда `cd ..` делает родительский каталог текущим.

Команда ls

Команда `ls` выводит содержимое каталога. Команда `ls` сначала выводит список всех файлов (не каталогов), перечисленных в командной строке, а затем выводит список всех файлов, находящихся в каталогах, перечисленных в командной строке. Если не указано ни одного файла, то по умолчанию аргументом назначается «.» (текущий каталог). Опция `-d` заставляет `ls` не считать аргументы-каталоги каталогами. Будут отображаться только файлы, которые не начинаются с «.» или все файлы, если задана опция `-a`.

Каждый список файлов (для файлов, которые не являются каталогами и для каждого каталога, содержащего список файлов) сортируется отдельно в алфавитной последовательности текущих региональных настроек (locale). Когда указана опция `-l`, то перед каждым списком выводится итоговая строка с общим размером всех файлов в списке, который измеряется в полу-килобайтах (512 байт). Результаты печатаются на стандартный вывод, по одному файлу на строку, если с помощью опции `-C` не задан многоколоночный вывод.

Команда ls

Для задания одноколончатого или многоколончатого режима вывода могут использоваться опции `-1` и `-C`, соответственно.

`ls [опции] [файл...]`

Опции POSIX: `[-CFRacdilqrut1] [--]`

Опции GNU (краткая форма):

`[1abcdfghiklmnopqrstuvwxABCFGHILNQRSUX] [-w cols] [-T cols]`

`[-l шаблон] [--full-time] [--show-control-chars] [--block-size=размер]`

`[--format={long, verbose, commas, across, vertical, single-column}]`

`[--sort={none, time, size, extension}] [--time={atime, access, use, ctime,`

`status}] [--color[={none, auto, always}]] [--help] [--version] [--]`

Опции POSIX

-C - напечатать список файлов в колонках с вертикальной сортировкой;

-F - Для каждого имени каталога добавлять суффикс `«/»`, для каждого имени FIFO – `«|»` и для каждого имени исполняемого файла `«*»`;

-R - включить рекурсивную выдачу списка каталогов;

-a - включать в список файлы с именем, начинающимся с `«.»`.

-c - использовать при сортировке (при задании опции `-t` или `-l`) время изменения состояния файла вместо времени последней модификации файла;

-d - выдавать имена каталогов, как будто они обычные файлы, вместо того, чтобы показывать их содержимое;

-i - предварять вывод для каждого файла его номером (номером inode);

Опции POSIX

-l - выдавать (в одноколоночном формате) тип файла, права доступа к файлу, количество ссылок на файл, имя владельца, имя группы, *размер* файла (в байтах), временной штамп и имя файла. По умолчанию, временной штамп является временем последней модификации; *опции* -с и -u позволяют выбрать два других временных штампа. Для файлов устройств поле *размера* обычно заменяется на старший и младший номера устройства.

-q - вместо непечатаемых символов в имени файла при выводе будут ставиться знаки вопроса. Эта опция включена по умолчанию при выводе на терминал;

-r - производить сортировку в обратном порядке;

-t - сортировать по показываемому временному штампу.

-u - использовать при сортировке (опция -t) или перечислении (опция -l) время последнего доступа к файлу вместо времени последней модификации файла;

-1 - выводить в формате одного столбца;

-- - завершает список опций.

Если стандартный вывод является терминалом, то вывод производится в многоколоночном формате (с вертикальной сортировкой).

Программа **dir** (также установленная под именем **d**) является эквивалентом «**ls -C -b**»; таким образом, файлы по умолчанию выдаются в многоколоночном формате (с вертикальной сортировкой). Программа **vdir** (также установленная под именем **v**) эквивалентна «**ls -l -b**»; таким образом, файлы по умолчанию выдаются в длинном формате.

Опции GNU

-1, --format=single-column - выводить один файл на строку.

Опции POSIX

Эта опция включается по умолчанию, если стандартный вывод не является терминалом;

-a, --all - вывести все файлы в каталогах, включая все файлы, начинающиеся с «.»;

-b, --escape, --quoting-style=escape - заменять неграфические символы в именах файлов, используя алфавитно-цифровые последовательности, а также последовательности с восьмеричными цифрами. Данная опция аналогична **-Q**, за исключением того, что имена файлов не заключаются в двойные кавычки;

-c, --time=ctime, --time=status - сортировать содержимое каталога в соответствии с временем изменения состояния файла (поле «ctime» в inode). Если с помощью *опции* **-l** задан длинный формат, то выдавать время изменения состояния файла вместо времени его модификации;

-d, --directory - выводить имена каталогов, как будто они обычные файлы, вместо того, чтобы показывать их содержимое;

-f - Не сортировать содержимое каталога; выдавать файлы в том порядке, в котором они записаны на диск. Эта опция также разрешает **-a** и **-U** и запрещает **-l, --color, -s, и -t**, если они были заданы перед **-f**;

-h, --human-readable - добавлять к каждому *размеру* файла букву *размера*, например, **M** для двоичных мегабайт;

-i, --inode - выдавать номер inode, также называемый номером индекса, каждого файла, слева от его имени. Этот номер однозначно идентифицирует каждый файл в каждой файловой системе;

-k, --kilobytes - если выдаются *размеры* файлов, то выдавать их в килобайтах;

Опции POSIX

-l, --format=long, --format=verbose - в дополнении к имени каждого файла, выводятся тип файла, права доступа к файлу, количество ссылок на файл, имя владельца, имя группы, *размер* файла в байтах и временной штамп (время последней модификации файла, если не задано другое). Для файлов с временем больше чем 6 месяцев назад или больше, чем 1 час в будущее, временной штамп содержит год вместо времени дня;

-m, --format=commas - выдавать список файлов в строку с максимально возможным количеством имен файлов на строку, разделяя имена запятой и пробелом;

-n, --numeric-uid-gid - выдавать числовые UID и GID вместо имен владельца и группы;

-o - применять длинный формат, но не показывать информацию о группе. Данная опция эквивалентна **--format=long --no-group**;

-p, --file-type, --indicator-style=file-type - добавлять символ, показывающий тип файла;

-q, --hide-control-chars - выдавать знаки вопроса вместо неграфических символов в имени файла. Эта опция включена по умолчанию.

-r, --reverse - сортировать содержимое каталога в обратном порядке;

-s, --size - выдавать *размер* каждого файла в блоках по 1024 байта слева от имени файла.

-t, --sort=time - сортировать по времени последней модификации (поле «mtime» в inode) вместо того, чтобы производить сортировку по алфавиту;

Опции POSIX

-u, --time=atime, --time=access, --time=use - сортировать по времени последнего доступа к файлу, вместо времени последней модификации (поле «atime» в inode). Если задан длинный формат вывода, выдавать время последнего доступа вместо времени последней модификации;

-v - производить сортировку в соответствии с номером версии файлов. При этом учитывается, что имена файлов часто содержат номера версий или порядковые номера;

-w, --width cols - назначает вывод на экран в *cols* колонок. По умолчанию, если это возможно, данное значение определяется из параметров драйвера терминала; в противном случае используется значение переменной окружения **COLUMNS**, если она установлена; в противном случае, по умолчанию, устанавливается 80;

-x, --format=across, --format=horizontal - выдавать список файлов в многоколоночном режиме, с сортировкой по горизонтали:

-A, --almost-all - выдавать все файлы, кроме «.» и «..»;

-B, --ignore-backups - не показывать файлы, которые заканчиваются на «~», если они не заданы в командной строке;

-C, --format=vertical - выдавать файлы в многоколоночном режиме, с сортировкой по вертикали. Опция устанавливается по умолчанию, если стандартный вывод является терминалом;

-D, --dired - при длинном формате (**-l**), после основного вывода, выдавать дополнительную строку вида: **//DIRED//** *BEG1* *END1* *BEG2* *END2*. Здесь *BEGn* и *ENDn* -- это целые числа, которые для каждой записи показывают байтовую позицию начала и конца каждого имени файла. Если каталоги отображаются рекурсивно (**-R**), то похожая строка выводится после каждого подкаталога: **//SUBDIRED//** *BEG1* *END1* ...;

Опции POSIX

-F, --classify, --indicator-style=classify - добавлять к каждому имени файла символ, показывающий его тип. Для обычных исполняемых файлов это «*». Для каталога добавляется «/», для FIFO – «|», для символических ссылок «@», для сокетов «=», для обычных файлов ничего не добавляется;

-G, --no-group - не отображать информацию о группе в длинном формате вывода.

-H, --si - делает то же, что и опция **-h**, но использует официальные единицы измерения. Для расчетов используется 1000 вместо 1024 и, таким образом, М - это 1000000, а не 1048576;

-I, --ignore=шаблон - не показывать файлы, имена которых совпадают с заданным *шаблоном* (*шаблон* - это не регулярное выражение), если только они не заданы в командной строке. Начальная «.» в имени файла не совпадает с символом «*», заданным в начале *шаблона*.

-L, --dereference - выдавать информацию о файлах, на которые указывают символические ссылки, вместо информации о самих символических ссылках;

-N, --literal - не заключать в кавычки имена файлов;

-Q, --quote-name, --quoting-style=c - заключать имена файлов в двойные кавычки и показывать неграфические символы как в языке C;

-R, --recursive - рекурсивно выдавать список содержимого всех каталогов;

-S, --sort=size - производить сортировку по *размеру* файла, вместо сортировки по алфавиту. Сначала будут показаны наибольшие файлы;

-T, --tabsize cols - назначить ширину табуляции в *cols* колонок. По умолчанию - 8. Если *cols* равно нулю, то табуляция не используется.

Опции POSIX

- U, --sort=none** - не производить сортировку; список файлов будет выдаваться в том порядке, в каком файлы записаны на диск (отличие между опциями **-U** и **-f** состоит в том, что данная опция не запрещает и не разрешает другие *опции*);
- X, --sort=extension** - производить сортировку в алфавитном порядке по расширениям файлов (символы после последней точки «.»). Файлы без расширений будут показаны первыми;
- block-size=size** - Выдает *размеры* в блоках по *size* байт;
- color[=when]** - задает цвет для различения типов файлов;
- full-time** - выдавать время в полном, а не в стандартном сокращенном варианте;
- quoting-style=word** - используйте стиль *word* для выбора режима квотинга имен файлов. Стиль *word* может принимать следующие значения:
 - **literal** - вывод имен файлов как есть. Это поведение **ls** по умолчанию;
 - **shell** - заключает имена файлов в одиночные кавычки для shell, если они содержат метасимволы или могут привести к неоднозначной выдаче.
 - **shell-always** - заключает имена файлов в одиночные кавычки для shell, даже если этого не требуется.
 - **c** - заключает имена файлов в двойные кавычки, как в языке C; это равносильно *опции* **-Q**.
- **escape** - как и для **-c**, за исключением того, что имя не заключается в двойные кавычки; эквивалентно *опции* **-b**.
- show-control-chars** - выдавать неграфические символы в именах файлов как есть.

Опции POSIX

Команда **mkdir**

В текущей директории можно создавать новые файлы и подкаталоги, которые будут являться подкаталогами текущей директории. Для создания одного или нескольких подкаталогов используется команда «**mkdir**» - создание каталога.

Синтаксис

mkdir [*опции*] *каталог...*

Опции POSIX: **[-p] [-m права] [--]**

Опции GNU (краткая форма): **[-p] [-m права] [--verbose] [--help] [--version] [--]**

Команда **mkdir** создает каталоги с заданными именами. По умолчанию права доступа к каталогам устанавливаются в 0777 ('a+gwx') за вычетом битов, установленных в **umask**.

ОПЦИИ

-m права, --mode=права - устанавливает *права* доступа к создаваемым каталогам. Эти права могут быть заданы либо в символьном виде, либо в виде восьмеричного числа.

-p, --parents - создает недостающие родительские каталоги для каждого указанного *каталога*. Права доступа к родительским каталогам устанавливаются в значение **umask**, логически сложенное с 'u+wx'. Аргументы, соответствующие существующим каталогам, игнорируются. Таким образом, если каталог /a существует, то '**mkdir /a**' выдаст ошибку, а '**mkdir -p /a**' – нет;

--verbose - выдавать сообщение для каждого создаваемого каталога. Это наиболее полезно с опцией **--parents**;

Команда **mkdir**

-- - Завершает список опций.

Вновь созданный каталог не будет пустым. Он будет содержать два элемента с символьными именами «.» и «..».

Необходимо обратить внимание на то, что любой каталог является файлом типа «d» с соответствующими правами доступа. Поэтому для создания нового каталога пользователь должен иметь право записи в родительский каталог.

Команда **rm**

Команда **rm** удаляет файлы или каталоги

Синтаксис

rm [*опции*] *файл...*

Опции POSIX: [-fiRr] [--]

Опции GNU (краткая форма): [-dfirvR] [--help] [--version] [--]

rm удаляет каждый заданный *файл*. По умолчанию каталоги не удаляются, но если заданы опции -г или -R, то будет удаляться все дерево каталогов ниже заданного каталога, включая и его самого без ограничения на глубину этого дерева. Будет выдано сообщение об ошибке, если последний компонент *файла* -- это «.» или «..» чтобы избежать неприятных сюрпризов при задании команды `rm -г .*' или ей подобных.

Если задана опция -i или файл является недоступным на запись и при этом стандартный вывод - это терминал и не задана опция -f, то **rm** выводит на стандартный вывод ошибок запрос на подтверждение удаления этого файла и читает ответ из стандартного ввода. Если ответ не утвердительный, то файл пропускается.

Команда **rm**

ОПЦИИ POSIX

-f - не запрашивать подтверждения операции, не выдавать диагностических сообщений, не возвращать код ошибочного завершения, если ошибки были вызваны несуществующими файлами;

-i - выводить запрос на подтверждение операции удаления (если заданы одновременно опции **-f** и **-i**, то срабатывает последняя указанная);

-r или **-R** - рекурсивное удаление дерева каталогов.

[--] - завершает список опций.

ОПЦИИ GNU

-d, --directory - удалять каталоги с помощью системного вызова `unlink` вместо `rm_dir`, и не требовать, чтобы каталог был пуст перед его разлинковкой. Работает только если у вас есть соответствующие привилегии. Поскольку разлинковка (разрыв связи) каталога приводит к тому, что все файлы в удаленном каталоге теряют ссылочную целостность, то будет благоразумно после этой операции проверить файловую систему командой `fsck`.

-f, --force - игнорировать несуществующие файлы и никогда не запрашивать подтверждение на удаления;

-i, --interactive - выдавать запрос на удаление каждого файла. Если ответ не утвердителен, то файл пропускается;

-r, -R, --recursive - рекурсивно удалять содержимое каталогов;

-v, --verbose - выдавать имя каждого файла перед его удалением.

Команда **rm**

Команда **rmdir**

Команда **rmdir** удаляет пустые каталоги. Если какой-либо из аргументов каталог не указывает на существующий пустой каталог, то будет выдано сообщение об ошибке.

Синтаксис

rmdir [*опции*] *каталог...*

Опции POSIX: [-p] [--]

Опции GNU (краткая форма): [-p] [--ignore-fail-on-non-empty] [--help] [--version] [--]

ОПЦИИ POSIX

-p - если *каталог* включает более чем один компонент пути, то удаляется каталог, затем убирается последний компонент пути и удаляется получившийся каталог и т. д. до тех пор, пока все компоненты не будут удалены. Таким образом, команда ***rmdir -p a/b/c*** эквивалентна последовательности команд ***rmdir a/b/c; rmdir a/b; rmdir a.***

[--] - завершает список опций.

ОПЦИИ GNU

--ignore-fail-on-non-empty - Обычно **rmdir** будет отказываться удалять непустые каталоги. Данная опция заставляет **rmdir** игнорировать ошибки при удалении каталога, если эти ошибки вызваны тем, что каталог не пуст.

-p, --parents - описана выше.

Команда **mv** (от англ. move) —используется для перемещения или переименования файлов.

Если в качестве аргументов заданы имена двух файлов, то имя первого файла будет изменено на имя второго: **mv file1**

Команда mv

file_1 - переименовывает файл.

Если последний аргумент является именем существующего каталога, то mv перемещает все заданные файлы в этот каталог: mv file ./dir/ перемещает «file» в «dir/file» относительно текущего каталога.

Если последний аргумент не является каталогом и задано более чем два файла, то выдается сообщение об ошибке.

Опции команды mv:

–f - не запрашивать подтверждения операций;

–i - выводить запрос на подтверждение операции, когда существует файл, в который происходит переименование или перемещение.

Команда cp

Команда cp (от англ. copy) копирует файлы или каталоги.

Если последний аргумент является существующим каталогом, то команда cp копирует каждый из заданных в качестве параметров команды файлов в этот каталог. Если задано только два имени файла, то команда копирует первый файл во второй.

Подробности применения команды можно получить, используя команду man cp.

Команда cat

Команда cat (от англ. concatenate) — последовательно выводит указанные файлы, таким образом, объединяя их в единый поток. Если вместо имени файла указывается «-», то читается стандартный ввод.

cat a.txt b.txt – содержимое файлов a.txt и b.txt выводится на экран;

cat a.txt b.txt > abc.txt - содержимое файлов a.txt и b.txt сохраняется в файле abc.txt.

Основы системного администрирования и сетевых технологий
