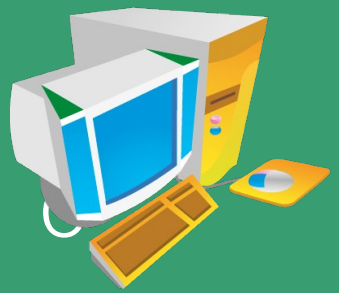
Гуляев Г.М.

Операционная система Linux и СПО

Занятие 4. Работа с файлами

Курс по переобучению на использование СПО



Пользователи и пароли

- Список всех пользователей (включая системных) находится в файле **/etc/passwd**

  $cat /etc/passwd (вывод таблицы на экран)

  username:pswd:uid:gid:uid comments:directory:shell

- Список всех групп (включая системные) находится в файле **/etc/group**

  $cat /etc/group (вывод таблицы на экран)

  groupname:pswd:gid:list

- uid - уникальный идентификатор пользователя
- gid - уникальный идентификатор группы
- Хэши паролей хранятся в файле **/etc/shadow**
- Каждый пользователь входит в группу и, как правило, не в одну (одна группа является основной)
- Команда **groups** [пользователь] выводит список групп в которые входит пользователь
- Команда **id** - список групп с кодами

Типы файлов в Linux

- **Для вывода списка файлов служит команда ls:**

  ls [имя] (без параметров - имя=.)

- **В Linux отсутствует атрибут скрытый, однако файлы или каталоги, имя которых начинается с точки многие команды «не замечают»:**

  ls -a [имя] (вывести полный список)
  ls -l [имя] (расширенный формат - в виде таблицы)

  george@george:~$ ls -l

  итого 17

  drwxr-xr-x 9 george george 4096 2011-11-09 17:29 Backup

  -rwxr-xr-x 1 george george 1299 2011-03-23 21:51 mkarch

  lrwxrwxrwx 1 george george 21 2010-11-07 17:00 mkdvd -> scripts/makedvd

- **Выведенная таблица содержит столбцы:**

  1. тип и разрешения (права), 2. количество ссылок

  3. имя пользователя, 4. имя группы

  5. занимаемый объем на диске (байт)

  6, 7. дата и время последнего изменения

  8. имя файла

Типы файлов в Linux

- **В Linux все является файлом: (каталоги, устройства, ...) Первый символ первого столбца указывает на тип файла:**

  - обычный файл, т.е. последовательность байт (текстовые документы, исполняемые программы и т.п.);

  d каталог, то есть именованный набор ссылок на другие файлы

  l символическая ссылка (symlink)

  b или c - устройство (блочное или символьное)

  p именованный канал (named pipe)

  s гнездо (socket)

- **Является ли файл исполняемым (то есть программой) или нет определяется правами пользователей (буква x в первом столбце вывода команды ls -l) на его исполнение, то есть одни смогут его запустить как программу, а другие нет.**

- **Если мы имеем права на запуск файла как программы, но он программой не является, то при запуске появится ошибка**

Права (разрешения) на файлы

- **У каждого файла в Linux имеется владелец (только один).**
- **9 символов (со 2 по 10) первого столбца (вывода ls -l) задают права доступа к файлу.**
- **Они делятся на три тройки, обозначающие права: владельца, членов его группы и всех остальных. Внутри каждой тройки может присутствовать или отсутствовать: право чтения (r), записи (w) и исполнения (x, от execute).**
- **1 тройка - для пользователя (владельца файла), 2 тройка - для группы пользователя-владельца, 3 тройка - для остальных (не входящих в эту группу).**
- **Пример: -rwxr-xr-x (тире убираем и получаем rwxrxrx). То есть владелец имеет все права: rwx (чтение, запись, исполнение), член его группы: rx (чтение, исполнение), остальные: rx (чтение, исполнение)**
- **Максимальные права: rwxrwxrwx. Права удобно представлять в двоичном виде: 1 - есть (r,w или x), 0 - нет. Тогда разрешения можно представить как 3 восьмиричных числа**

Права (разрешения) на файлы

- **rwx=111=7, rx=101=5, x=001=1, rwxrwxrwx=777, rwxrxx=751**

  (1=001, 2=010, 3=011, 4=100, 5=101, 6=110, 7=111)
- **Задание прав при помощи букв (rwx) называется символьным, а при помощи восьмиричных чисел абсолютным или двоичной маской.**
- **Для изменения прав доступа служит команда chmod, которая в качестве аргумента понимает как абсолютное, так и символьное задание прав.**
- **Примеры:**

  $chmod 640 readme (640=110 100 000=rwr)

  $ls -l readme

  -rw-r 1 george george 2556 2009-10-02 readme

  $chmod go+x readme

  $ls -l readme

  -rw-r-x--x 1 george george 2556 2009-10-02 readme

Права (разрешения) на файлы

- **В символьном варианте аргумента chmod явно указывается, кому какое право мы хотим добавить, отнять или присвоить:**

  chmod <категория действие набор_прав> < файл >

- **Категории пользователей:**

  u - владелец, g - группа, o - остальные, a - все (то же что ugo)

- **Действия:**

  + добавить, - отнять, = назначить

- **Права:**

  r - чтение, w - запись, x - исполнение, u - как у владельца,

  g - как у группы, o - как у остальных,

  s - право смены идентификатора пользователя или группы,

  t - бит прилипчивости (sticky bit)

- **Бит прилипчивости, установленный для каталога, приводит к тому, что удалять файлы из этого каталога может только владелец файла и владелец каталога (пример /tmp).**

Права (разрешения) на файлы

- **Команда ls -l вместо x может выводить буквы s, S у пользователя или группы и t, T у остальных:**

  s если бит смены идентификатора пользователя(группы)=1 и при этом x есть у владельца(группы)

  S если бит смены идентификатора пользователя(группы)=1 и при этом x отсутствует у владельца(группы)

  t если бит прилипчивости=1 (у остальных) и при этом x есть у остальных

  T если бит прилипчивости=1 и при этом x отсутствует у остальных

- **Можно присваивать все права, включая дополнительные биты, также в цифровом виде:**

  chmod 4755 readme

- **Здесь первой цифрой задается сумма из весов:**

  4 - бит смены идентификатора пользователя=1,

  2 - бит смены идентификатора группы=1,

  1 - бит прилипчивости=1.

Права (разрешения) на файлы

- **Файл можно передать другому владельцу или в другую группу командами:**

  chown [ключи] <новый пользователь>[:новая_группы] <файл> или

  chgrp [ключи] < новая_группа > <файл>

- **Каждому файлу соответствует индексный дескриптор (inode), содержащий параметры:**

  номер, тип файла, права доступа к файлу,

  количество связей (ссылок на файл в каталогах) файла,

  идентификатор пользователя и группы-владельца,

  размер файла в байтах, время последнего доступа к файлу,

  время последнего изменения файла и т. д.

- **Вывод номеров inode: команда ls -i**

- **Индексный дескриптор не содержит:**

  имени файла - оно содержится в блоках хранения данных каталога

  содержимого - оно размещено в блоках хранения данных файла

Жесткие и символические ссылки

- **Жесткая ссылка является просто другим именем для исходного файла. После создания такой ссылки (команда ln) ее невозможно отличить от исходного имени файла (один и тот же индексный дескриптор - inode).**
- **Создадим жесткую ссылку на файл readme и посмотрим, что изменилось в его свойствах:**

  $ln readme readme2

  $ls -l readme*

  -rw-r-x--x 2 george george 2556 2009-10-02 10:02 readme

  -rw-r-x--x 2 george george 2556 2009-10-02 10:02 readme2
- **Удаление файла по любому из его имен уменьшает на единицу количество ссылок, и окончательно файл будет удален только тогда, когда это количество станет равным нулю.**
- **Жесткую ссылку на файл можно создавать в любом каталоге, но обязательно на том же разделе диска (то есть в той же файловой системе), что и исходный файл.**

Жесткие и символические ссылки

- Команда **ls** с ключом -i выводит список с номерами inode
- Команда **stat** <имя_файла> [<имя_файла> ...] выдает информацию о файлах, включая индексный дескриптор (inode).
- Команда **find** позволяет находить жесткие ссылки на файл:

  find [каталог] -samefile имя_файла
- Символическая ссылка создается той же командой **ln** с ключом -s:

  $**ln** -s readme readme3

  $**ls** -l readme3

  lrwxrwxrwx 1 george george 16 2009-10-02 12:15 readme3 -> readme
- В поле имени файла после стрелки указано его настоящее имя. Полные права доступа у всех символических ссылок ничего не значат: возможность доступа определяется правами исходного файла.

Жесткие и символические ссылки

- **Файл-ссылка имеет ненулевую длину: в нем хранится абсолютное имя исходного файла. Команда вывода файл-ссылки на экран ничего не выводит:**

  $cat readme3

- **Прочитать на что ссылка можно так:**

  $readlink readme3

  readme

- **Символические ссылки на каталог создаются так же, как на обычный файл. Можно создавать и цепочку ссылок (ссылки на ссылки)**
- **Символическую ссылку на каталог можно использовать в пути к файлу, но вместо .. будет подставляться родительский каталог каталога-оригинала.**
- **Так, если в домашнем каталоге пользователя ivan есть ссылка link на домашний каталог пользователя den, то путь /home/ivan/link/.. эквивалентен не /home/ivan, а /home/den/.., то есть /home**

Шаблоны файлов

- Для того чтобы в именах файлов задавать не один файл а много существуют специальные символы *, ?, [], вместо которых командный интерпретатор перед выполнением команды подставляет реальные данные.
- * и ? играют такую же функцию, как и в DOS и Windows:

  * - любое число допустимых символов

  ? - один допустимый символ
- Примеры:

  ls *.* - все файлы в текущем каталоге

  ls * - все файлы в текущем каталоге и его всех подкаталогах
- В шаблонах также могут использоваться квадратные скобки []. В скобках задается диапазон подряд идущих символов:

  [a-z], [0-9], [A-F] и т.п

  или просто перечисляются символы:

  [abc], [49] и т.п
- Результатом является один символ из перечисленных

Шаблоны файлов

- **Например, шаблону [bar] соответствуют только символы a, b и r, но не c, B, bar или ab.**
- **Если после [в шаблоне стоит !, то результатом будет строка из одного символа не перечисленного между скобками (отрицание).**
- **Примеры:**

ls *.txt вывести имена файлов текущей директории с расширением txt

ls *[0-9].* вывести имена файлов текущей директории, в которых в имени последний символ - цифра

ls *[0-9] вывести имена файлов текущей директории, в которых в имени последний символ - цифра и отсутствует расширение

ls [a-c]??.* вывести имена файлов с любым расширением текущей директории, в которых первая буква имени равна a, b или c и имя содержит 3 символа.

Команды для работы с файлами

- **Создать пустой файл можно командой:**

  $ touch имя_файла

  Если файл существует, то эта утилита меняет время последнего изменения файла на текущее время, а если файла нет - она создает его.

- **Текстовые файлы можно создавать, вводя текст с консоли:**

  $ cat > имя_файла

  Далее можно вводить многострочный текст, а для окончания ввода и записи текста нажать Ctrl+D (символ конца файла).

- **Знак > здесь указывает на перенаправление ввода-вывода. В этом случае команда cat считает своими входными данными поток байтов, поступающий с клавиатуры, и выводит его в указанный файл.**

- **Если файл с указанным именем существует, то команда cat перепишет его. Чтобы добавить данные в конец существующего файла, следует вместо > использовать >> для перенаправления.**

Команды для работы с файлами

- **Удалить файл или каталог можно командой:**

  rm [ключи] имя

- **Некоторые ключи команды rm**

  -i : требовать подтверждения удаления для каждого удаляемого файла. Например, rm -i chernovik*, - будет запрос на подтверждение каждого удаляемого файла (ответы: y или n);

  -f : не запрашивать подтверждения и не выводить сообщений об ошибках, если указаны оба ключа -i и -f, то срабатывает последний указанный;

  -r : рекурсивное удаление каталога со всеми его подкаталогами. Непустой каталог можно удалить только так.

- **Создать каталог можно командой:**

  mkdir имя_каталога

- **Удалить пустой каталог можно командой:**

  rmdir имя_каталога

- **Для создания или удаления файла или подкаталога пользователю нужно иметь право записи (w) у каталога-родителя**

Команды для работы с файлами

- **Для копирования файлов служит команда:**

  cp [ключи] источник результат

  источник: файл(ы) или каталог, результат: файл(ы) или каталог

- **Ключей много, вот некоторые:**

  -i : требовать подтверждения при перезаписи файла;

  -f : не требовать подтверждения при перезаписи файла;

  -r : рекурсивное копирование каталога со всеми его подкаталогами;

  -d : копировать символические ссылки вместо файлов;

  -l : создавать жесткие ссылки вместо копирования;

  -s : создавать символические ссылки вместо копирования;

  -u : не переписывать существующий файл, если он изменен позже;

  -x : оставаться в пределах одной файловой системы.

- **Командой cat также можно копировать файлы:**

  cat источник > файл_результат

  Пример: cat *.txt > new_file.txt (слияние файлов)

Команды для работы с файлами

- **Команда для перемещения (переименовывания) файлов:**

  mv [ключи] источник результат

  источник: файл(ы) или каталог, результат: файл(ы) или каталог

- **Ключи аналогичны ключам команды cp:**

  -i : требовать подтверждения при перезаписи файла;

  -f : не требовать подтверждения при перезаписи файла;

  -r : рекурсивное перемещение каталога со всеми его подкаталогами;

Выполнить самостоятельно

1. Вызовите помощь по командам ls, cp, mv, rm, mkdir, rmdir при помощи ключа --help. Ознакомьтесь с информацией (запоминать все не обязательно)
2. В домашнем каталоге создайте директорию с вашим ФИО.
3. Выяснить разрешения (кто имеет какие права) на эту директорию.
4. Перейти в созданный вами новый каталог.
5. Командой touch создать в вашем каталоге файл myfile1.
6. Выяснить разрешения на этот файл.
7. Создать в вашем каталоге еще несколько файлов myfile2, myfile3, ..., myfile5, а также файлы hip, hop и help.
8. Создать в вашем каталоге поддиректорию mydir2.
9. Командой ls с параметром -F узнать, сколько файлов и директорий находится в вашем каталоге.
10. Вывести на экран только имена файлов hip и hop.
11. Скопировать в каталог mydir2, файлы, начинающиеся с букв myfile.
12. Используя специальные символы [], вывести на экран имена файлов myfile1-myfile4.

Выполнить самостоятельно

13. Используя команду chmod, для файла myfile1 дать

разрешение на запись в него для группы и других пользователей.

14. Создать при помощи cat текстовый файл hello, содержащий строку

echo Привет!

Добавить пользователю разрешение его исполнять как программу и запустить на выполнение.

15. Создать жесткую ссылку hi на созданный файл. Убедиться, что запуск hi приводит в тому же результату, что и запуск файла hello. Посмотреть количество жестких ссылок для обоих файлов. Удалить hi. Посмотреть количество жестких ссылок для hello.

16. Создать символическую ссылку hi на созданный файл. Убедиться, что запуск hi приводит в тому же результату, что и запуск файла hello.

17. Создать символическую ссылку privet на файл hi и запустить privet.

18. Проанализировать результат ls -l и определить какой файл не является символической ссылкой.

19. Создать символическую ссылку на какую-либо известную команду, например, cp и проверить ее работу.

20. Вернуться домой (cd) и удалить каталог с вашим ФИО командой rm

Вопросы для самоконтроля

1. Что представляет собой командный интерпретатор?
2. Какая разница между встроенными командами и внешними командами?
3. Какова структура команды для интерпретатора bash?
4. Что указывается в опциях команды?
5. Как можно получить информацию по использованию команды?
6. Чем отличается суперпользователь от обычного пользователя?
7. Что такое жесткая ссылка?
8. Что представляет собой символическая ссылка?
9. Можно ли создать символическую ссылку на отсутствующий файл?
10. Как определить, является ли файл символической ссылкой?
11. Как проследить цепочку символических связей?
12. Что такое абсолютное путь? Относительное путь?
13. Что такое права доступа к файлу? Можно ли их изменить?
14. Где хранится имя файла?
15. Что такое шаблон имени файла?
16. Как работает автозаполнение команд?
17. Как пользоваться историей команд?
18. Как пользоваться контекстным поиском команд?

Спасибо за внимание !

www.altailand.ru