

## Файловая структура системы GNU/LINUX

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 видам файлам однотипный. Такой подход обеспечивает независимость программы пользователя от особенностей ввода/вывода на конкретное внешнее устройство.

Файловая структура LINUX имеет иерархическую древовидную структуру. В корневом каталоге размещаются другие каталоги и файлы, включая 5 основных каталогов:

- bin - большинство выполняемых командных программ и shell-процедур;
- tmp - временные файлы; usr - каталоги пользователей;
- etc - административные утилиты и файлы;
- dev - специальные файлы, представляющие периферийные устройства.

Текущий каталог - это каталог, в котором в данный момент находится пользователь. При наличии прав доступа пользователь может перейти в любой 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): **ln <TARGET> <LINK>**, где <TARGET> - полное имя файла, с которым устанавливается связь; <LINK>- полное имя файла в новом каталоге, где будет использоваться эта связь. Новое имя может не отличаться от старого. Каждый файл может иметь несколько связей, то есть он может использоваться в разных каталогах под разными именами. Команда ln с аргументом -s создает символическую связь: **ln -s <TARGET> <LINK>**, здесь имя 2-го файла является именем символической связи. Символическая связь является особым видом файла, в котором хранится имя файла, на который эта связь ссылается. LINUX работает с символической связью не так, как с обычным файлом - например, при выводе на экран содержимого символической связи появятся данные файла, на который эта связь ссылается.

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:

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

Первый символ байта атрибутов определяет тип файла и может принимать следующие значения:

- обычный файл «-»;
- каталог «d»;
- символическая связь «l»;
- блок-ориентированный файл, который соответствует накопителям на магнитных дисках «b»;
- байт-ориентированный файл, который соответствует принтеру или терминалу «с».

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:

- (+) - добавить права доступа;
- (-) - 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:

- установка атрибутов чтения и записи для группы и чтения для всех остальных пользователей «`chmod g+rw,o+r`»;
- чтение атрибутов файла «`ls -l`»;
- отмена атрибута записи у остальных пользователей «`chmod o-w`»;
- создание/регистрация файла `letter` «`>letter`». Символ `>` используется как для переадресации, так и для создания файла;
- вывод содержимого файла «`cat`»;
- конкатенация файлов (объединение) «`cat <файл1> <файл2> > <файл3>`»;
- переименование файла «`mv <старое_имя> <новое_имя>`»;
- `mv` - перемещение файла в указанную директорию;
- `rm` - удаление файла;
- `cp` - копирование файла с переименованием;
- `mkdir` - создание каталога;
- `rm` - удаление каталога;

Запрос на вывод содержимого даётся командой `ls [acdfgilqrstv CFR]` .  
Значения аргументов:

- `l` – вывести всю информацию о файле;
- `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.

Переход в нижележащий каталог или каталог в котором содержится рабочий каталог реализуется командой “cd”. Если параметры не указаны, то происходит переход в домашний каталог пользователя. pwd - вывод имени текущего каталога; grep [-vcilns] [шаблон поиска] - поиск файлов с указанием или без указания шаблона.

Значение ключей-модификаторов:

- v – выводятся строки, не содержащие шаблон поиска;
- c – выводится число строк, содержащих или не содержащих шаблон;
- i – регистронезависимый поиск;
- l – выводятся только имена файлов, содержащие указанный шаблон;
- n – перенумерация выводимых строк;
- 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.
```

Файловая система ОС Ubuntu устроена таким образом, что все ресурсы представлены единообразно (в виде файлов). Такой подход позволяет обеспечить универсальный интерфейс доступа к любым ресурсам: от физических устройств до процессов, выполняющихся в системе.

Для обеспечения единообразного доступа к файлам их прежде всего необходимо классифицировать. В Ubuntu это сделано следующим образом:

- обычные (regular) файлы - текстовые, исполняемые, графические и пр. файлы, создаваемые пользователями и прикладными программами;
- каталоги (directories) - именованные группы файлов и вложенных каталогов;
- файлы устройств (devices) - соответствуют присутствующим в системе реальным (жесткие диски, принтеры, мыши, ЦП и т. д.) и т.н. псевдоустройствам (например, /dev/null). Файлы устройств представляют символьные

(последовательного доступа) и блочные (произвольного доступа) устройства. К первым относят, например, параллельные и последовательные порты, ко вторым - жесткие диски;

- специальные файлы - сокеты (sockets) и именованные каналы (named pipes), которые предназначены для обмена информацией между процессами;
- символичные ссылки (symlinks) - именованные указатели на физические файлы (аналог ярлыков ОС Windows), содержащие только путь к некоторому файлу. Символичные ссылки могут указывать на файлы, хранящиеся как локальных, так и в сетевых каталогах.

Символичные ссылки (или "мягкие") не нужно путать с "жесткими", которые указывают на inode файла. Inode (идентификатор узла) - это уникальный числовой идентификатор узла (файла или каталога) файловой системы, по которому и осуществляется доступ к нему. Символическое имя файла ориентировано на пользовательское восприятие, так как для человека-оператора проще запомнить осмысленные имена файлов (например: report.txt, myfoto.jpg и т.п.), чем абстрактные числовые значения.

### **Каталоги в ОС семейства GNU/Linux**

Все файлы упорядочены по каталогам. Структура и назначение каждого из каталогов, созданных на этапе установке, predetermined, хотя и могут быть изменены суперпользователем.

Файловая система имеет иерархическую структуру и начинается от корневого каталога (/). Его подкаталогами являются:

- /bin - исполняемые файлы общего назначения;
- /boot - содержит образ загружаемого ядра;
- /dev - файлы устройств;
- /etc - конфигурационные файлы общего пользования;
- /home - домашние каталоги пользователей, включая программы и файлы личных предпочтений;
- /lib - общесистемные библиотеки;
- /mnt - каталог монтирования внешних файловых систем;
- /proc - виртуальная файловая система для чтения информации о процессах;
- /root - домашний каталог суперпользователя;
- /sbin - программы системного администрирования;
- /tmp - каталог для хранения временной информации;
- /usr - каталог пользовательских прикладных программ со всеми их исполнимыми и конфигурационными файлами;
- /var - каталог для хранения часто изменяющихся файлов, например, спулера печати, различных лог-файлов, почтовых сообщений и т.п.

/lost+found - каталог для нарушенных фрагментов файлов, обнаруженных в результате проверки файловой системы после сбоя.

Такая структура типична для большинства дистрибутивов Linux, но ОС может иметь и другие каталоги, например,

/opt — для дополнительных компонентов,

/selinux - расширение системы безопасности и т.п.

### **Именованние файлов и каталогов**

Файловая система Ubuntu поддерживает "длинные" имена, содержащие символы латиницы, национальных алфавитов, знаки пунктуации и спецсимволы. Абсолютно запрещенными к использованию в имени являются прямой и обратный слэши (/ и \). Максимальное количество символов в имени - 255. Понятие "расширения файла" в Unix-системах отсутствует как таковое, поэтому в имени может быть несколько частей, разделенных точками. Все имена - регистрозависимые .

Файлы и каталоги, названия которых начинаются с точки (dot-файлы), являются аналогами "скрытых" файлов MS-DOS, то есть не отображаются при просмотре содержимого файловой системы.

Для быстрого доступа к файлам в оболочке имеются несколько переменных окружения, хранящих соответствующие пути.

Это, например, переменная \$HOME, в которой содержится пути к домашнему каталогу текущего пользователя. Таким образом, выполнение команд

```
[usr1@localhost var]$ cd /home/usr1
```

и

```
[usr1@localhost var]$ cd $HOME
```

приведут к одному результату - переходу в домашний каталог пользователя usr1. Более того, в оболочке определен псевдоним для домашнего каталога - символ ~ (тильда) можно использовать аналогично \$HOME. Например:

```
[usr1@localhost var]$ cd ~
```

```
[usr1@localhost ~]$ pwd
```

```
/home/usr1
```

```
[usr1@localhost var]$
```

### **Команды управления файловой системой**

Управление файловой системой производится при помощи различных команд, реализующих операции по созданию, чтению, копированию, переименованию/перемещению, изменению и удалению файлов и каталогов.

Основными командами для выполнения файловых операций являются: `pwd`, `ls`, `cp`, `mv`, `dir`, `rm`, `cd`, `rmdir`, `mkdir`, `ln`. Информация об их назначении и параметрах доступна в справках `man` и `info`.

Файлы в Ubuntu могут быть созданы как результаты работы прикладных программ и иметь при этом определенный формат (например, графические файлы, созданные редактором GIMP) или могут быть созданы пользователем путем ввода информации с клавиатуры, например:

```
aag@stilo:~> cat > f1
```

Hello, world! // нажатие Ctrl+D завершает ввод команд

```
aag@stilo:~>
```

Также файлы могут быть созданы путем перенаправления вывода команды со стандартного потока:

```
[root@localhost aag]# echo "Hello, World!" > f1
```

В данных примерах символ ">" - это команда перенаправления стандартных потоков ввода/вывода, встроенная в оболочку. В первом случае она получает информацию со стандартного потока ввода (клавиатура) и по окончании ввода (Ctrl+D) отправляет ее в файл. Во втором - принимает строку, переданную командой `echo`, и также отправляет ее в файл. Если файл отсутствует, то он будет создан, если имеется, то будет перезаписан.

Для добавления информации в файл следует использовать команду ">>".

### **Рекомендуемый порядок выполнения работы**

1. Ознакомиться с файловой структурой ОС Ubuntu, найти и увидеть содержимое каталогов, соответствующих NFS. Изучить функциональность и синтаксис записи команд манипуляций с файлами: `rm`, `rmlink`, `touch`, `mkdir`, `cp`, `mv`, `>`, `<`, `>>`, `<<`.
2. Используя команды, создать два текстовых файла.
3. Объединить созданные в п.№ 27 файл и его содержимое просмотреть на экране.
4. Создать директорию с другим именем и переместить в нее созданные ранее файлы.
5. Вывести полную информацию обо всех файлах и проанализировать уровни доступа, дешифровав маски доступа.

6. Добавить для всех трех файлов право выполнения членам группы и остальным пользователям.
7. Просмотреть атрибуты файлов.
8. Создать еще один каталог с другим именем.
9. Установить дополнительную связь объединенного файла с новым каталогом, но под другим именем.
10. Создать символическую связь.
11. Сделать текущим новый каталог и вывести на экран расширенный список информации о его файлах.
12. Произвести поиск заданной последовательности символов в файлах текущей директории и получить перечень соответствующих файлов. (одной командой)
13. Получить информацию об активных процессах и именах других пользователей.
14. Вывести на экран список файлов текущего каталога в краткой и расширенной форме.
15. Изменить текущий каталог на корневой «/».
16. Записать в файл \$HOME/filelist.lst список каталогов в каталоге /.
17. Вернуться в домашний каталог и вывести рекурсивный список всех (в т.ч. и скрытых) файлов и каталогов.
18. В домашнем каталоге (\$) создать подкаталоги src, dst и temp.
19. В каталоге src создать текстовый файл f1 произвольного содержания.
20. В каталог src скопировать файлы user.txt, root.txt и stud.txt. Проверить, все ли файлы удалось скопировать.
21. В каталоге dst создать "жесткие" ссылки на все файлы из каталога src .
22. В домашнем каталоге создать "мягкие" ссылки на файлы из каталога src .
23. Вывести рекурсивно расширенную информацию о содержимом домашнего каталога. Обратить внимание на поле размера для созданных ранее файлов и ссылок.
24. Из домашнего каталога выполнить команды: cat /src/f1; cat /dst/f1; cat /f1. Запомнить результаты выполнения команд.
25. Переместить файл f1 из каталога src в каталог temp и повторить п.49.
26. Удалить файл f1 и повторить п.49.