

Лабораторная работа

Системные каталоги. Основные команды для работы с файлами

Цели работы:

- изучить назначение основных системных каталогов
- приобрести навыки работы с базовыми командами для управления файлами и каталогами в операционных системах семейства GNU/Linux
- научиться использовать встроенную справочную систему man

Теоретические сведения

Файловая система Linux

В операционных системах семейства GNU/Linux все системные компоненты, от физических устройств до работающих процессов и конфигурационных настроек, представлены в виде файлов. Этот фундаментальный принцип Unix-подобных систем, известный как *"Everything is a file"*, обеспечивает единообразный способ взаимодействия с системой через стандартные файловые операции.

Преимущества подхода:

- **унификация:** для взаимодействия с разными сущностями (устройствами, процессами, конфигурациями) используются стандартные инструменты командной строки;
- **модульность:** возможность замены компонентов без изменения системы: альтернативные shell, выбор драйверов через /sys/class, простое подключение новых устройств (автоматическое создание в /dev);
- **гибкость:** механизмы перенаправления ввода-вывода (>, <, >>) и конвейеры (|) позволяют реализовать обмен данными между файлами, процессами и устройствами; использование символьных ссылок для создания альтернативных путей;
- **прозрачность:** все системные данные доступны для чтения/модификации через текстовые файлы, что упрощает отладку и настройку, не требуя обязательного наличия GUI или специализированных утилит;
- **отказоустойчивость:** возможность восстановления через ручное редактирование конфигурационных файлов, анализ текстовых логов;
- **автоматизация:** простое создание скриптов для анализа логов, мониторинга и конфигурирования системы;
- **предсказуемость:** структура и поведение документированы и стандартизированы.

В операционных системах семейства GNU/Linux используется моноиерархическая файловая система, соответствующая стандарту FHS (Filesystem Hierarchy Standard) с единой точкой монтирования - корневым каталогом (/). Все компоненты файловой системы, включая физические устройства, сетевые ресурсы и виртуальные файловые системы,

интегрируются в единое дерево посредством механизма монтирования в подкаталоги корня.

Применение стандарта обеспечивает единую структуру каталогов (/bin, /etc, /home, /var и др.) во всех дистрибутивах, чёткое разделение файлов по назначению (программы, настройки, временные данные), совместимость между разными версиями операционных системах семейства GNU/Linux.

Таблица 1. Основные системные каталоги

Каталог	Назначение
/	корневой каталог (основа всей файловой системы)
/bin	основные исполняемые файлы (команды: ls, cp, mv)
/sbin	основные системные утилиты
/etc	конфигурационные файлы системы
/home	домашние каталоги пользователей
/var	изменяемые файлы (логи, кэш, базы данных)
/tmp	временные файлы (очищаются при перезагрузке)
/usr	пользовательские программы, библиотеки и документация
/dev	файлы устройств (жесткие диски, терминалы и т.д.)
/proc	виртуальная файловая система (информация о процессах и ядре)
/lib	основные системные библиотеки, необходимые для загрузки системы и запуска команд, содержащихся в корневой файловой системе.

Системные каталоги содержат критически важные компоненты операционной системы, их удаление или модификация может привести к нарушению работоспособности системы, потере важных данных и необходимости полной переустановки системы. При необходимости внесения изменений в файлы в /etc делайте это с осторожностью – используйте sudo и делайте бэкапы.

Основные команды для работы с файлами

1. Навигация по файловой системе

pwd (print working directory) - показывает полный путь к текущему каталогу.

Синтаксис команды:

```
pwd
```

Пример:

```
pwd # Вывод: /home/user
```

ls (list directory contents) - выводит список файлов и подкаталогов.

Синтаксис команды:

```
ls [опции] [каталог]
```

Основные опции:

- l – подробный вывод (права, владелец, размер, дата)
- a – показать скрытые файлы (начинающиеся с `.`)
- h – размеры в удобочитаемом формате (KB, MB)

Примеры:

```
ls # Список файлов в текущей папке
```

`ls -l /etc` # Подробный список каталога /etc

cd (change directory) – переход в указанный каталог. Если каталог не указан, используется значение переменной окружения \$HOME.

`.` - указывает на текущий каталог

`..` - указывает на родительский каталог (на уровень выше).

Синтаксис команды:

`cd [каталог]`

Примеры:

`cd /var/log` # перейти в /var/log

`cd ~` # вернуться в домашний каталог

2. Работа с файлами и каталогами

mkdir (make directory) - создание нового каталога

Синтаксис команды:

`mkdir [опции] имя_каталога`

Основные опции:

`-p` - создать все родительские каталоги, если их нет (например, `mkdir -p dir1/dir2`)

`-m` - установить права доступа при создании (например, `mkdir -m 755 mydir`).

Пример:

`mkdir new_folder` # создать каталог "new_folder"

touch - создание пустого файла или обновление времени модификации существующего.

Примеры:

`touch file.txt` # создать file.txt (если нет)

`touch file1 file2` # создать несколько файлов

cp (copy) - копирование файлов и каталогов

Синтаксис команды

`cp [опции] файл1 [файл2 ...] целевой_файл`

Команда `cp` копирует файл1 в целевой_файл. Если целевой_файл является каталогом, то файл1, файл2, ... копируются в него под своими именами. Если целевой_файл существует и не является каталогом, его старое содержимое теряется. Время последней модификации целевого_файла, а также время последнего доступа к исходным файлам устанавливается равным времени, когда выполняется копирование.

Основные опции:

`-r` - рекурсивное копирование (для каталогов)

`-i` - запрос подтверждения перед перезаписью

`-v` - вывод информации о процессе копирования (verbose)

`-p` - сохранить атрибуты (файла (права, владельца, время))

Пример:

`cp file.txt /backup/` # Копировать file.txt в /backup/

mv (move) - перемещение или переименование файлов.

Синтаксис команды:

`mv файл1 [файл2 ...] целевой_файл`

Команда `mv` перемещает (переименовывает) файл1 в целевой_файл. Имя

файла1 не должно совпадать с именем целевого файла. Если целевой_файл является каталогом, то файл1, файл2, ..., перемещаются в него под своими именами. По умолчанию подтверждение запрашивается только при отсутствии прав на запись.

Работа с каталогами. Переименовать или переместить можно только каталоги, находящиеся в одной файловой системе, при этом содержимое каталога сохраняется.

Полезные опции:

- i - интерактивный режим (подтверждение перед перезаписью)
- f - принудительное выполнение (не спрашивать подтверждения)
- v - вывод информации о выполняемых действиях

Примеры:

```
mv old.txt new.txt      # переименовать old.txt в new.txt
mv *.pdf ~/Documents/  # Перемещение всех файлов .pdf
```

rm (remove) - удаление файлов и каталогов

Синтаксис команды:

```
rm [опции] файл_или_каталог...
```

Основные опции:

- f - принудительное удаление без запросов подтверждения (игнорирует отсутствие файла).
- i - интерактивный режим: запрашивает подтверждение перед удалением каждого файла.
- r или -R - рекурсивное удаление (для каталогов: удаляет содержимое вместе с самой папкой).
- v - вывод информации о каждом удаляемом файле.

Если заданное имя было последней ссылкой на файл, то файл уничтожается. Для удаления пользователь должен обладать правом записи в каталог.

Удалённые с помощью rm файлы стираются почти безвозвратно (восстановление возможно только с помощью специальных утилит и при определённых условиях). Команда `rm -rf /` – уничтожает всю систему. В современных Linux-системах эта команда автоматически блокируется для защиты от случайного уничтожения системы.

Вместо rm можно использовать trash-cli (аналог корзины), или создать алиас для безопасного удаления. Для этого добавьте в ~/.bashrc строку:

```
alias rm='rm -i' # всегда запрашивать подтверждение
```

Опция -i часто используется совместно с -r. По команде: `rm -ir dirname` запрашивается подтверждение: `directory dirname: ?`. При положительном ответе запрашиваются подтверждения на удаление всех содержащихся в каталоге файлов, а затем подтверждение на удаление самого каталога.

rmdir (remove directory) - удаляет только пустые директории. Безопасная альтернатива rm -r, так как не может удалить директории с файлами, директории с поддиректориями и отдельные файлы.

Синтаксис команды:

```
rmdir [-p] [-s] каталог ...
```

Убедитесь, что текущий рабочий каталог не является подкаталогом удаляемой директории. Для удаления каталогов нужно иметь те же права доступа, что и в случае удаления обычных файлов [см. rm].

Опции:

- p рекурсивно удаляет цепочку пустых родительских директорий.
- s не выводит сообщение, выдаваемое при наличии опции -p.

ln (link) - создает жёсткие и символические ссылки

Синтаксис команды:

```
ln [опции] исходный_файл ссылка # жёсткая ссылка
ln -s исходный_файл ссылка # символическая ссылка
```

Жёсткая ссылка - это дополнительная запись в каталоге, имеющая тот же индексный дескриптор (inode), что и исходный файл. Inode содержит всю системную информацию о файле (права доступа, имя владельца, временные метки, адресную информацию), за исключением его имени. Все жёсткие ссылки, связанные с одним inode, равнозначны: при удалении любой из них счётчик ссылок в inode уменьшается на единицу, сами данные остаются доступными до тех пор, пока счётчик не обнулится.

Жесткую ссылку нельзя создать на каталог (для предотвращения циклических зависимостей) и между разными файловыми системами, так как inode уникальны в пределах только одного раздела диска.

Символическая ссылка - это специальный файл, содержащий текстовый путь к целевому объекту. В отличие от жёсткой ссылки, символическая ссылка имеет собственный inode и независимые атрибуты, включая права доступа. Это означает, что даже при полном доступе к целевому файлу сама ссылка может быть защищена от чтения или выполнения. При этом важно понимать, что стандартные операции (например, chmod), применённые к символической ссылке, воздействуют на целевой файл, а не на саму ссылку. Символические ссылки работают с каталогами и могут указывать на объекты в других файловых системах, но становятся неработоспособными при перемещении или удалении целевого файла.

При создании ссылки в случае, если указанное имя ссылки уже существует в файловой системе, команда запросит подтверждение на перезапись (в интерактивном режиме). Чтобы подавить этот запрос и принудительно перезаписать ссылку, используйте опцию -f.

Примеры:

```
# Создаём жёсткую ссылку report_backup на report.txt:
ln report.txt report_backup
# Создаём символическую ссылку в домашнем каталоге:
ln -s /var/log/app.log ~/current_log
```

cat (concatenate) - просмотр содержимого файлов, объединение файлов, создание новых файлов.

Синтаксис команды:

```
cat [опции] [файл1] [файл2] ... [файлN]
```

Основные возможности команды:

- просмотр содержимого файлов

```
cat file.txt # выводит содержимое file.txt
```

- объединение (конкатенация) файлов

```
cat file1 file2 > combined.txt # объединение файлов
```

- создание нового файла

```
cat > newfile # создаёт файл (завершить Ctrl+D)
```

- добавление (дозапись) в конец файла:

```
cat >> existing.txt
```

3. Просмотр и поиск

less / more - постраничный просмотр файлов

Синтаксис команды:

```
less [опции] [файл]
```

```
more [опции] [файл]
```

Пример:

```
less /var/log/syslog # просмотреть лог с прокруткой
```

grep (Global Regular Expression Print) - поиск текста в файлах или потоке данных по заданному шаблону (регулярному выражению или точному совпадению).

Синтаксис команды:

```
grep [опции] "шаблон" [файл ...],
```

где «шаблон» - текст или регулярное выражение для поиска, «файл» - один или несколько файлов для анализа (если файл не указан, читается стандартный ввод).

Примеры:

```
# Поиск слова "error" в файле:
```

```
grep "error" /var/log/syslog
```

```
# Поиск в выводе другой команды:
```

```
cat file.txt | grep "warning"
```

find - мощный инструмент для поиска файлов и каталогов в Linux, позволяет осуществить поиск по имени, размеру, времени изменения и другим параметрам.

Примеры:

```
find /home -name "*.txt" # Найти все .txt-файлы
```

```
find /var/log -size +10M # Файлы больше 10 МБ
```

Современные альтернативы find и grep:

fd - быстрый аналог find с подсветкой и простым синтаксисом.

ripgrep (rg) - ускоренный grep с поддержкой .gitignore и подсветкой.

Установка: `apt install fd-find ripgrep`

4. Справочная информация и получение помощи

man (manual) - встроенная справочная система Linux, которая предоставляет официальную документацию по командам, системным вызовам, конфигурационным файлам и другим аспектам операционной системы, разделенную на категории:

1. пользовательские команды (например, ls, cp)
2. системные вызовы (функции ядра, например, open())
3. библиотечные функции (например, printf)
5. форматы файлов (например, /etc/passwd)
8. администрирование (команды для root, например, mount).

Man - самый быстрый и доступный способ узнать синтаксис команды, существующие опции, примеры использования, связанные команды.

Синтаксис команды:

```
man [раздел] [команда]
```

Пример:

```
man ls          # Показать документацию по команде ls
man 1 printf    # Документация по команде printf
man 3 printf    # Документация по функции printf (ЯП C)
```

Всегда указывайте раздел man для избежания ошибок. Если термин относится к файлу (например, /etc/fstab), используйте man 5, если это команда - man 1 или man 8.

Полезные опции:

```
-k [ключевое_слово] - поиск по всем мануалам
-f [команда] - показать краткое описание.
```

Пример:

```
# найти все страницы, связанные с копированием файла
man -k "copy file"
```

Альтернативы man:

info - более подробная документация.

--help (-h) - краткая справка (работает не для всех команд).

tldr - упрощённая альтернатива man, которая показывает краткие примеры использования команд. Подходит для быстрого поиска распространённых вариантов использования.

Порядок выполнения работы

Задание 1. Изучение структуры файловой системы

1. Откройте терминал.
2. Выполните команду pwd, зафиксируйте результат.
3. Перейдите в корневой каталог (cd /), выведите список его содержимого (ls -l).
4. Изучите назначение и содержимое каталогов /bin, /sbin, /etc, /dev, /proc, /home, /var, /user, /temp, /lib (для анализа используйте ls -l, file и man hier). Приведите в отчете назначение основных файлов и подкаталогов.
5. Найдите PID своего shell (с помощью echo \$\$) и изучите его каталог в /proc/[PID]/.

Задание 2. Работа с файлами и каталогами.

1. Войдите в свой домашний каталог (cd ~).
2. Создайте следующую структуру каталогов и файлов:
 - в домашнем каталоге создайте каталог Ваша_фамилия (латиницей).
 - перейдите в созданный каталог и создайте в нем каталог practice_1
 - внутри каталога practice_1 создайте каталог catalog1, каталог catalog2,

файл file1 (например, используя команду echo; запишите в него свою фамилию, текущее время и дату). Перейдите в каталог catalog2.

- внутри каталога catalog2 создайте файлы file2 и file3 и каталог catalog3.
- внутри каталога catalog3 создайте файл file4, запишите в него вывод из man для команды по варианту:
1: ls, 2: cd, 3: pwd, 4: mkdir, 5: rmdir, 6: cp, 7: mv, 8: rm, 9: touch, 10: cat, 11: less, 12: more, 13: head, 14: tail, 15: grep, 16: find, 17: chmod, 18: chown, 19: ps, 20: top, 21: kill, 22: df, 23: du, 24: free, 25: uname, 26: history, 27: alias, 28: echo, 29: date, 30: man.
- создайте жесткую ссылку на файл file1, жесткую ссылку на каталог catalog2.
- создайте в каталоге practice_1 символическую ссылку s_link на файл file4.

3. Проконтролируйте структуру созданных вами каталогов, просмотрите содержимое файлов. Проверьте работоспособность ссылок.

Структуру каталогов в консольном режиме можно просмотреть с помощью команды tree. Утилита выводит дерево каталогов в отформатированном виде. При необходимости установите ее:

```
sudo apt-get install tree
```

Для вывода дерева каталогов команда может иметь следующий вид:

```
tree -a practice_1
```

В отчете по работе приведите скриншоты, демонстрирующие выполнение всех пунктов задания.

Дополнительные задания

Задание 1. Используя команды ls, du и grep в каталоге /var/log

- найдите 5 самых больших файлов и выведите их отсортированными по размеру
- определите, какие файлы были изменены за последние 2 дня.
- извлеките все строки, содержащие ошибки ("error" или "fail"), из /var/log/syslog

Задание 2. Анализ файлов системных каталогов

- Проанализируйте файлы в каталоге /etc. Составьте список всех файлов в /etc, которые имеют расширение .conf или содержат слово "network". Для каждого найденного файла выведите его размер, дату последнего изменения и владельца.
- Из /proc/cpuinfo извлеките информацию о модели процессора и выведите количество ядер.
- Из /proc/meminfo определите общий объем ОЗУ, объем свободной памяти.

Задание 3. Сравнение содержимого системных каталогов

- Найдите команды (исполняемые файлы), которые присутствуют в каталогах /bin и /usr/bin
- Определите, какие версии команд новее (используйте ls -l для сравнения дат модификации)

- Проверьте, есть ли символические ссылки между каталогами /bin и /usr/bin
- Сравните типы и количество библиотек в /lib и /usr/lib, определите присутствующие в обоих каталогах библиотеки.

Контрольные вопросы

1. В чем разница между операцией > и >> ?
2. Для чего используется каталог /home?
3. Какую информацию можно получить из каталога /proc?
4. Какой командой можно перейти из одного каталога в терминале в другой? Приведите пример команды перехода из каталога пользователя usr1 в корневой каталог
5. В чем разница между файлом и каталогом?
6. Как можно создать новый каталог?
7. Какая команда выдает справку по другой команде?
8. Как можно получить документацию по системному вызову exit? По команде exit?
9. В чем разница между жесткими и символическими ссылками?
10. Что произойдет с жесткой ссылкой, созданной командой: *ln file1 link1*, если удалить файл file1: *rm file1*?