

==== справка по основным командам linux ====

=====

Команды Linux для управления файлами

1. ls

Утилита для просмотра содержимого каталогов. По умолчанию показывает текущий каталог. Если в параметрах передать путь, то она перечислит содержимое того каталога. Полезные опции -l (List) и -a (All). Первая форматирует вывод в виде списка с более подробной информацией, а вторая включает показ скрытых файлов.

Примеры:

```
ls # файлы в текущем каталоге
ls ../some_folder # вернуться на уровень назад и заглянуть в папку some_folder
ls /home/user/some_folder # заглянуть в папку по абсолютному пути
ls -l # показать дополнительную информацию о файлах
```

2. cat

Печатает содержимое файла, переданного в параметре, в стандартный вывод. Если передать несколько файлов, команда склеит их. Также можно перенаправить вывод в еще один файл, с помощью символа '>'. Если нужно вывести только определенное количество строк используйте опцию -n (Number).

Примеры:

```
cat file.txt
cat *.txt # открыть все файлы в текущей папке с расширением .txt
cat in.txt > out.txt # записать из in.txt в out.txt
```

3. cd

Позволяет перейти из текущего каталога, в указанный. Если запустить без параметров - возвращает в домашний каталог. Вызов с двумя точками возвращает на уровень вверх относительно текущего каталога. Вызов с тире (cd -) возвращает к предыдущему каталогу.

Примеры:

```
cd /home/username/folder # абсолютный путь
cd # в домашнюю папку
cd ../../ # на 2 уровня назад
```

4. pwd

Печатает на экран текущий каталог. Это может быть полезно, если ваша командная строка Linux не выводит такую информацию, а также особенно полезно в Bash программировании, для получения ссылки на каталог в котором выполняется скрипт.

5. mkdir

Создание новых каталогов. Наиболее удобная опция - p (Parents), позволяет создать всю структуру подкаталогов одной командой, даже если они еще не существуют.

Примеры:

```
mkdir some_folder
mkdir folder/some_folder # создаст some_folder только если существует folder (иначе
ошибка)
mkdir -p folder/some_folder # создаст some_folder и folder, если последний не существует
```

6. cp

Копирование файлов и каталогов. Она не копирует каталоги рекурсивно по умолчанию, поэтому не забудьте добавить опцию -r (Recursive) или -a (Archive). Последняя включает режим сохранения атрибутов, владельца и временного штампа, в дополнение к рекурсивному копированию.

Примеры:

```
cp file.txt /home/username/file_copy.txt      # скопирует файл
cp -r /home/somepath/somedir /home/some_other_path  # скопирует папку
```

7. mv

Перемещение или переименование файлов и каталогов. Примечательно, что переименование и перемещение - это одна и та же операция. Переименование - это перемещение файла в ту же папку с другим именем.

8. rm

Удаляет файлы и папки. Очень полезная команда Linux, с помощью нее вы можете убрать весь беспорядок. Однако будьте осторожны при ее использовании. Хотя и для того чтобы повредить систему вам нужно серьезно постараться, вы можете удалить собственные важные файлы. Rm не удаляет файлы в корзину, из которой потом все можно будет восстановить. Так что будьте осторожны, чтобы потом не говорили: "rm съела мою курсовую". Все действия необратимы. Если нужно рекурсивное удаление, используйте опцию -r.

Примеры:

```
rm some_file.txt
rm -rf some_dir      # папки удалятся только с -r опцией
```

9. ln

Создает жесткие или символические ссылки на файлы. Символические или программные ссылки - это что-то похожее на ярлыки в Windows. Они предоставляют удобный способ доступа к определенному файлу. Символические ссылки указывают на файл, но не имеют никаких метаданных. Жесткие ссылки в отличие от символических указывают на физический адрес области диска, где хранятся данные файла.

10. find

Поиск в файловой системе, файлов и папок. Это очень гибкая и мощная команда Linux не только из-за своих возможностей поиска, но и благодаря возможности выполнять произвольные команды для найденных файлов.

Примеры:

```
find -name "filename"      # искать filename начиная с текущей папки
find /home/ -name "filename"  # искать везде
find -name "some*.txt"      # искать файлы, имена которых начинаются на some с расширением txt
```

Linux команды консоли для работы с текстом

11. more / less

Это две простенькие команды терминала, для просмотра длинных текстов, которые не вмещаются на одном экране. Представьте себе очень длинный вывод команды. Или вы вызвали cat для просмотра файла и вашему эмулятору терминала потребовалось несколько секунд, чтобы

прокрутить весь текст. Если ваш терминал не поддерживает прокрутки, вы можете сделать это с помощью less. Less новее, чем more и поддерживает больше опций, поэтому использовать more нет причин.

12. head / tail

Еще одна пара, но здесь у каждой команды своя область применения. Head выводит несколько первых строк из файла (голова), а tail выдает несколько последних строк (хвост). По умолчанию каждая утилита выводит десять строк. Но это можно изменить с помощью опции -n. Еще один полезный параметр -f. Это сокращение от Follow (следовать), утилита постоянно выводит изменения в файле на экран. Например, если вы хотите следить за лог файлом, вместо того чтобы постоянно открывать и закрывать его используйте tail -nf.

13. grep

Grep как и другие инструменты Linux делает одно действие, но делает его хорошо. Она ищет текст по шаблону. По умолчанию она принимает стандартный ввод, но вы можете искать в файлах. Шаблон может быть строкой, или регулярным выражением. Она может вывести как совпадающие, так и несовпадающие строки и их контекст. Каждый раз, когда вы выполняете команду, которая выдает очень много информации, не нужно анализировать все вручную, пусть grep делает свою магию.

Примеры:

grep "word" file.txt	# искать строки со словом word в файле
grep "w.rd" file.txt > blah.txt	# нечеткий поиск и запись результата в blah.txt

14. sort

Сортировка строк текста по различным критериям. Наиболее полезные: -n (Numeric) - по числовому значению, и -r (Reverse), которая переворачивает вывод. Это может быть полезно для сортировки вывода du. Например, если хотите отсортировать файлы по размеру, просто соедините эти команды.

15. wc

Утилита командной строки Linux для подсчета количества слов, строк, байт и символов.

16. diff

Показывает различия между двумя файлами, в построчном сравнении. Причем выводятся только строки, в которых обнаружены отличия. Измененные строки отмечаются символом "с", удаленные - "d", а новые - "a".

Кстати, я написал еще одну подробную статью, в которой описан именно просмотр содержимого текстового файла в Linux с помощью терминала.

Команды Linux для управления процессами

17. kill / xkill / pkill / killall

Все они служат для завершения процессов. Но они принимают различные параметры для идентификации процессов. Kill нужен PID процесса, xkill - достаточно кликнуть по окну, чтобы закрыть его, killall и pkill принимают имя процесса. Используйте ту, которая удобна в определенной ситуации.

18. top / htop

Обе команды похожи, обе отображают процессы, и могут быть использованы как консольные системные мониторы. Я рекомендую установить htop, если в вашем дистрибутиве он не поставляется по умолчанию, так как это намного улучшенная версия top. Вы сможете не только просматривать, но и контролировать процессы через его интерактивный интерфейс.

=====

Команды Linux окружения пользователя

19. su / sudo

Su и sudo - это два способа выполнить одну и ту же задачу - запустить программу от имени другого пользователя. В зависимости от вашего дистрибутива, вы, наверное, используете одну или другую. Но работают обе. Разница в том, что su переключает вас на другого пользователя, а sudo только выполняет команду от его имени. Поэтому использование sudo будет наиболее безопасным вариантом работы.

=====

Linux команды для просмотра документации

20. man / whatis

Команда man открывает руководство по определенной команде. Для всех основных команд Linux есть man страницы. Whatis какие разделы руководств есть для данной команды.