**архиватор TAR**

### **tar** - наиболее распространенный архиватор, используемый в Linux-системах. Сам по себе tar не является архиватором в привычном понимании этого слова, т.к. он самостоятельно не использует сжатие. В то же время, многие архиваторы (например, [Gzip](http://en.wikipedia.org/wiki/Gzip) или [bzip2](http://en.wikipedia.org/wiki/bzip2)) не умеют сжимать несколько файлов, а работают только с одним файлом или входным потоком. Поэтому чаще всего эти программы используются вместе. tar создает несжатый архив, в который помещаются выбранные файлы и каталоги, при этом сохраняя некоторые их атрибуты (такие как права доступа). После этого полученный файл \*.tar сжимается архиватором, например, gzip. Вот почему архивы обычно имеют расширение .tar.gz или .tar.bz2 (для архиваторов gzip и bzip2 соответственно)

Мы можем архивироваться со сжатием и без. Что это значит? Без сжатия мы просто сделаем архив, в который наш архиватор запихнет некоторые файлы, которые нам нужны.

Но если применим специальный ключ, то запустится не только программа архивации, но и запустится процесс сжатия файлов.

Давайте заархивируем некоторые файлы, но прежде мы их создадим:

date > /tmp/1.txt

ps -ef > /tmp/2.txt

теперь зайдем в эту папку:

cd /tmp

И попробуем заархивировать эти файлы, но укажем, куда мы сразу запишем наш новый архив (в папку opt, например):

tar -czf /opt/arhive.tar.gz \*.txt

Что мы тут сделали? Для начала мы вызвали архиватор tar. Потом ключами указали, что мы будем делать. Напомню, чтобы вызвать подсказку по tar нам нужно в терминале написать:

tar –help (два минуса)

И как мы видим, тут мы указываем:

* с - означает, что мы создаем архив
* z - означает, что мы ужимаем файлы при помощи gzip
* f - указываем имя файла архива

Дальше мы указываем путь, в который попадет наш архив. В данном случае - это папка opt. ВАЖНО! Предварительно нужно находиться в папке с файлами, которые мы будем архивировать и ужимать.

Дальше, через пробел говорим архиватору, чтобы тот нашел в папке все файлы с расширением txt и добавил их в архив.

Теперь давайте попробуем разархивировать наши файлы в другое место и проверить.

Для разархивирования нам понадобится тот же архиватор tar:

tar -xzf /opt/archive.tar.gz -C /tmp/1

Ругается. Не понимает нашей папки 1 в папке /tmp

Создадим папку 1:

mkdir /tmp/1

Перезапускаем команду разархивации.

Что мы сделали?

1. вызвали наш tar
2. указали ключи:

* x - извлекаем
* z - используем gzip (так как мы видим, что файл не только в архиве, но и ужат)
* f - работаем с файлами

1. Указываем папку, из которой извлекаем наш архив и какой файл разархивируем
2. используем ключ -С (большое, чтобы указать другое месторасположение)
3. указываем саму папку, куда хотим разархивировать файлы.

теперь проверим наш новосозданный архив:

ls -la /opt/

и проверим наши файлы в новой папке после разархивации:

ls /tmp/1

cat /tmp/1/2.txt

Командная оболочка Bash позволяет искать файлы и папки на компьютере или сервере с помощью нескольких утилит, таких как `find`, `locate`, `grep`.

## Команда `find`

Команда `find` позволяет искать файлы и каталоги по различным критериям, таким как имя, тип, размер, дата изменения. Это очень полезная утилита, если нужно найти, переместить, переименовать или удалить множество файлов на компьютере или сервере.

Синтаксис команды `find` выглядит следующим образом:

```

find [path] [expression]

```

где `path` - это путь к каталогу, в котором нужно искать файлы, а `expression` - это условия, которым должны удовлетворять файлы.

### Поиск файлов по имени

Для поиска файлов по имени вы можете использовать флаг `-name`. Например, если вы хотите найти все файлы с именем `file.txt`, выполните следующую команду:

```

find /path/to/search -name "file.txt"

```

где `/path/to/search` - это путь к каталогу, в котором нужно искать файлы.

Вы можете также использовать шаблоны и регулярные выражения, для более сложного поиска файлов.

Например, следующая команда найдет все файлы с расширением `.txt` в текущем каталоге:

```

find . -name "\*.txt"

```

### Поиск файлов по типу и размеру

Вы можете использовать флаг `-type` для поиска файлов по типу. Например, следующая команда найдет все директории в текущей директории:

```

find . -type d

```

А следующая команда найдет все файлы в текущей директории, размер которых больше 1 МБ:

```

find . -type f -size +1M

```

### Поиск файлов по дате изменения

Вы можете использовать флаг `-mtime` для поиска файлов по дате изменения. Например, следующая команда найдет все файлы в текущей директории, которые изменены более 7 дней назад:

```

find . -type f -mtime +7

```

Вы также можете использовать `+` для поиска файлов, которые изменены более n дней назад, и `-` для поиска файлов, которые изменены менее n дней назад.

### Поиск файлов и выполнение действия

Команда `find` также позволяет выполнять определенные действия с найденными файлами. Например, чтобы найти все файлы с расширением `.txt` и удалить их, вы можете выполнить:

```

find . -name "\*.txt" -exec rm {} \;

```

где `{}` заменяется на имя найденного файла, а `;` указывает, что команда завершена.

## Заключение

В этой статье мы рассмотрели основы использования команды `find` для поиска файлов и каталогов в Linux. Эта утилита довольно мощная и гибкая, поэтому может быть очень полезной для выполнения различных задач в Linux. Надеюсь, эта статья помогла вам понять, как использовать команду `find` в своей работе с Linux.

++++++++++

1. Какие оболочки доступны в Linux, и как переключиться на другую оболочку?

Ответ: в Linux доступны различные оболочки, такие как bash, zsh, ksh и другие. Чтобы переключиться на другую оболочку, можно использовать команду `chsh -s /path/to/shell`, где /path/to/shell - путь к исполняемому файлу оболочки.

2. Как вывести список файлов в директории?

Ответ: Для вывода списка файлов в директории нужно использовать команду `ls`. Например, `ls /home/user/`.

3. Как узнать текущую директорию?

Ответ: Для узнавания текущей директории можно использовать команду `pwd`.

rm

Ответ: Для создания файла можно использовать команду `touch`. Например, `touch file.txt` создаст файл с именем `file.txt`.

5. Как удалить директорию и ее содержимое?

Ответ: Для удаления директории и ее содержимого нужно использовать команду `rm -r directory\_name`. Например, `rm -r /home/user/directory\_name`.

6. Как включить скрытые файлы и директории при выводе списка?

Ответ: Чтобы включить скрытые файлы и директории при выводе списка, нужно использовать опцию `-a` команды `ls`. Например, `ls -a`.

7. Как переместить (переименовать) файл?

Ответ: Для перемеще(переименования) файла можно использовать команду `mv`. Например, `mv file.txt new\_file.txt` переместит файл `file.txt` и переименует его в `new\_file.txt`.

8. Как скопировать файл?

Ответ: Для копирования файла в Linux нужно использовать команду `cp`. Например, `cp file.txt new\_file.txt` скопирует файл `file.txt` и создаст новый файл с именем `new\_file.txt`.

9. Как архивировать файлы или директории?

Ответ: Для архивирования файлов и директорий в Linux можно использовать команду `tar`. Например, `tar -cvf archive.tar file1 file2 directory` архивирует файлы `file1` и `file2` и директорию `directory` в архив `archive.tar`.

10. Как распаковать архив?

Ответ: Для распаковки архива в Linux можно использовать команду `tar`. Например, `tar -xvf archive.tar` распакует архив `archive.tar`.

11. Как создать символическую ссылку?

Ответ: Для создания символической ссылки в Linux используется команда `ln -s`. Например, `ln -s /path/to/file link` создаст символическую ссылку `link` на файл `/path/to/file`.

12. Как запустить команду в фоновом режиме?

Ответ: Для запуска команды в фоновом режиме в Linux нужно использовать символ `&` в конце команды. Например, `command &` запустит команду в фоновом режиме.

13. Как прервать выполнение команды в терминале?

Ответ: Для прерывания выполнения команды в терминале в Linux нужно использовать сочетание клавиш `Ctrl + C`.

14. Как просмотреть содержимое файла?

Ответ: Для просмотра содержимого файла в Linux можно использовать команду `cat` или `less`. Например, `cat file.txt`.

15. Как посмотреть первые строки файла?

Ответ: Для просмотра первых строк файла в Linux можно использовать команду `head`. Например, `head file.txt` выведет первые 10 строк файла.

16. Как посмотреть последние строки файла?

Ответ: Для просмотра последних строк файла в Linux можно использовать команду `tail`. Например, `tail file.txt` выведет последние 10 строк файла.

17. Как выполнить команду от имени другого пользователя?

Ответ: Для выполнения команды от имени другого пользователя в Linux можно использовать команду `su`. Например, `su - username` позволит войти в систему от имени пользователя `username`.

18. Как создать нового пользователя?

Ответ: Для создания нового пользователя в Linux нужно использовать команду `useradd`. Например, `useradd username` создаст пользователя с именем `username`.

19. Как изменить пароль пользователя?

Ответ: Для изменения пароля пользователя в Linux можно использовать команду `passwd`. Например, `passwd username` позволит изменить пароль для пользователя `username`.

20. Как получить помощь по команде?

Ответ: Для получения помощи по команде в Linux можно использовать команду `man`. Например, `man ls`.

21. Как установить программу из репозитория?

Ответ: Для установки программы из репозитория в Linux нужно использовать менеджер пакетов определенного дистрибутива. Например, для установки программы в Ubuntu нужно использовать команду `apt-get install program\_name`.

22. Как запустить программу из терминала?

Ответ: Для запуска программы из терминала в Linux нужно использовать команду, которая запускает исполняемый файл программы. Например, `./program\_name`.

23. Как написать скрипт на bash?

Ответ: Для написания скрипта на bash нужно создать текстовый файл с расширением `.sh` и добавить нужные команды. Например, `nano script.sh`.

24. Как запустить скрипт на bash?

Ответ: Для запуска скрипта на bash нужно использовать команду `bash` и указать путь к скрипту. Например, `bash /path/to/script.sh`.

25. Как создать переменную?

Ответ: Для создания переменной в Linux нужно использовать знак `=`. Например, `variable="value"`.