

# Копирование файлов с помощью rsync

ОПУБЛИКОВАНО: 04.08.2014 | АВТОР: НИКОЛАЙ



rsync в основном предназначен для удаленной синхронизации. rsync используется для выполнения операций резервного копирования в UNIX / Linux.

rsync — это утилита, которая используется для синхронизации файлов и директорий двух различных локаций. Считается как одна из самых эффективных. Резервные копии можно создавать как на локальном сервере так и на удаленном.

## Некоторые важные возможности rsync

**Скорость:** В первый раз, rsync реплицирует полностью весь контент между источниками и каталогами назначения. В дальнейшем, rsync перемещает только измененные блоки или байты в назначенную локацию, причем делает это действительно быстро. Также имеется возможность ограничить скорость синхронизации.

**Безопасность:** rsync позволяет шифровать информацию, используя ssh протокол в процессе перемещения данных.

**Меньшая пропускная способность:** rsync использует блочное сжатие и распаковку данных на приемной и отправляющей стороне в указанном порядке, возможность «докачки» файлов. Таким образом пропускная способность будет использоваться rsync-ом всегда гораздо меньше по сравнению с остальными протоколами передачи данных.

Если имеется доступ по **ssh** к серверу, это наиболее простой способ для осуществления синхронизации.

**Rsync по ssh** — это частный случай использования, общий вид команды выглядит так:

```
rsync [ОПЦИИ] [ПОЛЬЗОВАТЕЛЬ@]СЕРВЕР:ИСТОЧНИК... [ПОЛУЧАТЕЛЬ]  
Pull (копирование файлов с удаленной системы на локальную)
```

```
rsync [ОПЦИИ] ИСТОЧНИК 1 [ИСТОЧНИК N]  
[ПОЛЬЗОВАТЕЛЬ@]СЕРВЕР:ПОЛУЧАТЕЛЬ
```

Push (копирование файлов с локальной машины на удаленную)  
Обязательным параметром является «ИСТОЧНИК», их может быть несколько.

Рассмотрим несколько примеров:

### Примеры:

Скопируется на сервер 192.168.1.1 **содержимое** папок /backup/file1/ /backup/file2/ в папку backup

```
# rsync -zavP /backup/file1/ /backup/file2/  
user@192.168.1.1:/backup
```

В этом примере уже скопируются сами папки

```
# rsync -zavP /backup/file1/ /backup/file2/  
user@192.168.1.1:/backup
```

**Как ограничить скорость передачи файлов? Нужно поставить опцию --bwlimit :**

```
# rsync -zavP --bwlimit=100 /backup/file1/ user@192.168.1.1:/backup
```

## Как синхронизировать файлы с проверкой по контрольной сумме?

```
# rsync -czavP /backup/file1/ user@192.168.1.1:/backup
```

## Как скопировать файлы с удаленной машины на локальную?

```
# rsync -zavP user@192.168.1.1:/backup /backup/file1/
```

## Как исключить файл или директорию при копировании?

Исключаем директорию temp файл test.txt и все файлы с расширением tgz расположенные в /home/data

```
# rsync -azVP --exclude temp/ --exclude test.txt --exclude *.tgz /home/data/ user@192.168.1.1:/backup
```

## Как удалить файлы на приемнике, если их не существует на источнике?

```
# rsync -azVP --exclude /temp/ --exclude test.txt --exclude *.tgz -  
-force --delete /home/data/ user@192.168.1.1:/backup
```

Исключаем директорию /temp файл test.txt и все файлы с расширением tgz расположенные в /home/data

## Как использовать нестандартный порт ssh при копировании rsync?

```
# rsync -zavP '-e ssh -p 12345' /var/lib/ root@192.168.1.1:/backup
```

```
# rsync --progress -avz -e ssh /home/ root@adminunix.ru:/home/
```

Копирует с локального хоста содержимое папки home, на хост adminunix.ru в папку home. Для аутентификации используется учетная запись root. В случае обрыва соединения во время копирования, достаточно просто повторить команду и копирование начнется с того места, когда произошел обрыв соединения.

```
rsync --progress --bwlimit=10 -avz -e ssh /home  
root@adminunix.ru:/home/
```

Небольшое изменение. Удаляем «/» после папки home и у нас уже копируется сама папка вместе со всем содержимым. Так же ограничиваем скорость копирования до 10кБ/с, что бы не забивать интернет канал.

Опции:

**-a, --archive** режим архива, при использовании заменяет несколько ключей (-rlptgoD)

— n отладочный режим

**-t, --times** — обновлять время модификации файла на приёмной стороне. При отсутствии этой опции или **-a**становиться неэффективной оптимизация передачи по времени изменения файлов

**-r** рекурсивно. (Копирует все файлы, включая вложенные каталоги)

**-l** при копировании сохраняет символические ссылки

**-p** сохраняет права на файлы

— t сохраняет время изменения файлов

— g сохраняет группу файлов

— o сохраняет владельца файлов (работает только из под root)

— D сохраняет файлы устройств и специальные файлы

— P, отображение прогресса при копировании;

— q, не выводит сообщения об ошибках;

— c, --checksum проверка файлов по контрольной сумме, а по размеру и дате модификации. Дополнительная нагрузка на

- процессор, сильно увеличивает время синхронизации;
- size-only – использование для сверки только размер файлов
- ignore-errors, продолжение копирования и удаления даже после появления ошибок;
- max-delete, максимальное число удаляемых за один раз файлов и каталогов;
- files-from=FILE, задается список файлов и директорий для копирования в файл;
- numeric-ids – не транслировать имена владельца и группы в цифровые UID и GID, оставить на удалённой стороне номера как есть
- R – использовать относительные пути при создании символических ссылок
- A, --acls – сохранение списков ACL
- H, --hard-links – сохранение жестких ссылок
- S, --sparse – оптимизировать [urlspan][разреженные файлы](#)[/urlspan]
- x, --one-file-system – не выходить за пределы текущей точки монтирования
- u, update пропускать обновление файлов, которые новее исходных
- h, human-readable вывод цифр в читаемом виде (Кб, Мб, Гб)
- v verbose подробные вывод операций (отладочной информации)
- z сжатие файлов при копировании
- compress-level — уровень сжатия
- e ssh использовать при копировании ssh
- progress показывает прогресс выполнения копирования
- bwlimit=KBPS ограничивает скорость передачи файлов (Kbit/s)
- del – удалить файлы в папке назначения которых нет в источнике (точная копия)
- --delete-after удалить после. Если в основном месте был удален какой-то файл, или каталог, то после синхронизации в backup сервере, в каталоге он тоже будет удален
- password-file Путь, где находится файл с паролем.

Перед тем как запустить скрипт на рабочей машине, опробуйте всё в отладочном режиме, для этого используется ключ *-n*. В этом случае, rsync не будет менять или удалять файлы, но покажет, весь ход работы.

```
rsync -a --progress /home/ /mnt/export/
```

Копирует внутри одного компьютера содержимое папки *home*, в папку */mnt/export/*. Подобное копирование применяется когда необходимо сохранить все права на перемещаемые файлы.