

Unix & Linux Stack Exchange - это сайт вопросов и ответов для пользователей Linux, FreeBSD и других Unix-подобных операционных систем. Регистрация занимает всего минуту.

Любой может задать вопрос



Кто-нибудь может ответить

Зарегистрируйтесь, чтобы присоединиться к этому сообществу

Лучшие ответы оцениваются и попадают в топ

# UNIX & LINUX

## Сравните два zip-файла на предмет различий

Задано 5 лет и 7 месяцев назад    Изменено 10 месяцев назад    Просмотрено 39 тысяч раз



27



команда `diff` сравнивает, чтобы увидеть любую разницу между двумя файлами. Можно ли использовать то же самое для сравнения двух zip-файлов, т. е. Есть ли какая-либо разница в данных, таких как количество и т.д. В отдельных файлах в заархивированных файлах?

[разница](#)



Поделиться    Улучшите этот вопрос    Подписаться

спрошено 29 июня 2018 г. в 17:14



UnixPhile

371    1    3    3

Важен ли для вас порядок, в котором файлы включаются в zip-архив? – [guido](#) 29 июня 2018 г. в 17:25

порядок не имеет значения, так как сравниваются правильные файлы – [UnixPhile](#) 29 июня 2018 в 17:27

### 5 Ответов

Отсортировано по:

Highest score (default)



29



Вам придется распаковать их (если только в память), чтобы сравнить их. Классный способ, который я видел для этого с `diff`, - это:

```
diff -y <(unzip -l file1.zip) <(unzip -l file2.zip)
```



Это покажет вам, содержатся ли какие-либо файлы в одном, а не в другом



Поделиться Улучшите этот ответ Подписаться

ответил 29 июня 2018 г. в 17:26



Jaken551

565 3 9

4 Я думал о том же, но добавил `-qq1` вместо `-l` для подавления некоторого шума и сортировки по имени файла в конце | `sort -k4` –[guido](#) 29 июня 2018 г. в 17:29

1 отличные предложения @guido! –[Jaken551](#) 29 июня 2018 в 17:31

2 здесь сравнивается количество файлов. Но как насчет содержимого внутри файлов? – [UnixPhile](#) 29 июня 2018 в 17:42

Я бы разархивировал zip-файлы и создал два массива с файлами. Затем я бы использовал что-то вроде `str` для сравнения n-го элемента `arr1` с n-м элементом `arr2` –[Jaken551](#) 29 июня 2018 в 17:47

2 @UnixPhile `diff -y --suppress-common-lines -W 333 <(unzip -lqq file1.zip | sort -k 4) <(unzip -lqq /file2.zip | sort -k 4)` подавит одинаковые записи и покажет пропущенные / дополнительные файлы, а также файлы, отличающиеся размером или меткой времени. `-W` примерно такой ширины, должно быть достаточно большим для `loooong /path /to /files`. Сравнение по содержимому потребует больше усилий, дайте мне знать, если это действительно требуется. –[Tagwint](#) 29 июня 2018 в 18:14



7



Я опубликовал более подробное объяснение [в разделе "различать файлы внутри zip без их извлечения"](#) но если вы хотите сравнить *содержимое* файлов внутри zipfile и игнорировать все метаданные (в частности, временные метки), тогда вам следует запустить:

```
diff \
  <(unzip -vqq file1.zip | awk '{ $2=""; $3=""; $4=""; $5=""; $6=""; print }' | sort -k3) \
  <(unzip -vqq file2.zip | awk '{ $2=""; $3=""; $4=""; $5=""; $6=""; print }' | sort -k3)
```

Поделиться Улучшите этот ответ Подписаться

ответил 9 апреля 2020 г. в 3:50



Ян

253 3 7



1



Одним из вариантов сравнения ZIP-файлов и каталогов является использование "zipstr", как упоминалось в другом посте.

"zipstr" работает до тех пор, пока ZIP-архивы используют ту же структуру каталогов.

Если вам нужно сравнить 2 ZIP-архива с одинаковыми файлами, но в одном архиве файлы содержатся в дополнительном подкаталоге, "zipstr" помечает все файлы как измененные, что может быть проблемой. Таким образом, "zipstr" можно использовать только для проверки, является ли содержимое точно таким же.

"folderdiff" (<https://github.com/ssh-mitm/folderdiff>) был создан, потому что в некоторых случаях использования требуется сравнить резервную копию вашего веб-приложения с надежным источником.

Например, если вы хотите найти измененные файлы (например, возможные бэкдоры), вы можете использовать "folderdiff" для этой задачи, и его можно использовать с ZIP-архивами и папками.

Пример:

Wordpress использует ZIP-архив, где файлы в архиве хранятся в папке с именем "wordpress". В этом примере файлы извлекаются в "/ var / www /", но без вложенной папки wordpress.

"folderdiff" может игнорировать папку "wordpress /" из установочного архива с аргументом --prefix и перечисляет только "webshell.php" и "index.php" как разные файлы.

```
$ folderdiff wordpress-6.0.3-de_AT.zip backup.zip --prefix wordpress/  
===== Added =====  
+ webshell.php  
===== Modified =====  
* index.php
```

"zipstr" не может сравнить файлы из-за другого пути, используемого в архиве.

Раскрытие информации: я автор "folderdiff"

Поделиться Улучшите этот ответ Подписаться

ответил 6 марта 2023 г. в 14:37



Манфред Кайзер

141 4



0



Вы могли бы попробовать следующее:

1. Создайте репозиторий с какой-либо системой контроля версий (например, Git)
2. Распакуйте первый zip-файл
3. Зафиксируйте текущее содержимое репозитория
4. Очистите содержимое репозитория (за исключением его метаданных, например, папки .git)
5. Распакуйте второй zip-файл
6. Запустите diff (например, git diff)

Поделиться Улучшите этот ответ Подписаться

ответил 21 октября 2022 г. в 0:31



Соломон Учко

147 1 9



Правильный способ - контрольные суммы.

-2

```
diff <(md5sum file1.zip | cut -f1 -d ' ') <(md5sum file2.zip | cut -f1 -d ' ')
```



Поделиться Улучшите этот ответ Подписаться

ответил 17 июля 2021 г. в 21:14



Raimo

11 3



- 1 Это не работает, так как если вы добавите одни и те же файлы в два zip-файла, но временные метки файлов будут разными, тогда хэш файла не будет совпадать. –[Дэвид Руссель](#) 8 июня 2022 г. в 14:36

Если file1.zip и file2.zip не используют одинаковую степень сжатия и имеют другие отличающиеся параметры, результат будет недействительным. При сравнении содержимого zip-файлов необходимо распаковать файлы, прежде чем использовать md5sum для несжатого содержимого. –[Biary](#) 26 октября 2022 г. в 15:13



**Очень активный вопрос.** Заработайте 10 репутации (не считая бонуса за ассоциацию), чтобы ответить на этот вопрос. Требование репутации помогает защитить этот вопрос от спама и действий, не позволяющих ответить.