



BUP BACKUP [OPCIONES]



INDICE DOCUMENTO

[SERVICIO DE INICIO - I](#)
[Script.sh](#)

COPIAS DE SEGURIDAD BUP



NOMBRE

Bup index

imprime y/o actualiza el índice del sistema de archivos bup



SINOPSIS

```
bup index <-p|-m|-s|-u|--clear|--check> [-H] [-l] [-x] [--fake-valid] [--no-check-device]
[--fake-invalid] [-f indexfile] [--exclude path] [--exclude-from filename]
[--exclude-rx pattern] [--exclude-rx-from filename] [-v] <paths...>
```



DESCRIPCIÓN

El índice bup manipula el índice del sistema de archivos, que es una caché de rutas absolutas y sus metadatos (atributos, hashes SHA1, etc.). Su función

es similar a la del índice git(1), y el índice predeterminado se encuentra en

\$BUP_DIR/bupindex.

Crear una copia de seguridad en bup consta de dos pasos: actualizar el índice con bup index y, a continuación, realizar una copia de seguridad de los archivos (o de un subconjunto de ellos) con bup save. Esta separación se debe a las siguientes razones:

- 1 Hay más de una manera de generar una lista de archivos que necesitan copia de seguridad. Por ejemplo, podría usar inotify(7). o dnotify(7).
- 2 Incluso si realiza copias de seguridad de archivos en varios destinos (para mayor redundancia), los nombres de archivo, los atributos y los hashes serán los mismos cada vez. Por lo tanto, puede evitar la molestia de tener que regenerar repetidamente la lista de archivos para cada conjunto de copia de seguridad.
- 3 Es posible que desee utilizar los datos rastreados por el índice bup para otros fines (como acelerar otros programas que necesitan la misma información).





NOTAS

Por el momento, bup ignorará los atributos de Linux (cf. `chattr(1)` y `lsattr(1)`) En algunos sistemas (cualquier sistema bigendian donde `sizeof(long) < sizeof(int)`). Esto se debe a que el kernel de Linux y FUSE discrepan actualmente sobre el tipo de los argumentos de la llamada al sistema `attr`, por lo que en sistemas bigendian no es posible obtener los resultados sin el riesgo de corrupción de la pila (<http://lwn.net/Articles/575846/>).

En estas situaciones, bup imprimirá una advertencia la primera vez que los atributos de Linux sean relevantes durante cualquier operación de indexación/guardado/restauración. bup se adapta a la resolución de la marca de tiempo del sistema de archivos esperada en el peor de los casos, actualmente un segundo; por ejemplo, VFAT, ext2, ext3, ext4 pequeño, etc. Dado que bup no puede conocer la resolución de la marca de tiempo del sistema de archivos y podría estar recorriendo varios sistemas de archivos durante una ejecución, siempre asume que la resolución podría no ser mejor que un segundo.

En la práctica, esto significa que las actualizaciones de índice son algo imprecisas, por lo que bup save podría ocasionalmente registrar cambios inesperados en el sistema de archivos. Esto se debe a que, durante una actualización de índice, si bup encuentra una ruta cuyas marcas de tiempo reales son más recientes que un segundo antes del inicio de la actualización, bup establecerá las marcas de tiempo de índice para esa ruta (`mtime` y `ctime`) exactamente un segundo antes de la ejecución, lo que limita esos valores.

Esto garantiza que ningún cambio posterior en esas rutas pueda generar marcas de tiempo idénticas a las del índice. Si eso fuera posible, bup podría pasar por alto las modificaciones.

Puede ver el efecto de este comportamiento en este ejemplo (suponga que transcurre menos de un segundo entre la creación del archivo inicial y la primera ejecución del índice):

```
$ touch src/1 src/2
# A "sleep 1" here would avoid the unexpected save.
$ bup index src
$ bup save -n src src # Saves 1 and 2.
$ date > src/1
$ bup index src
$ date > src/2 # Not indexed.
$ bup save -n src src # But src/2 is saved anyway.
```

Estrictamente hablando, bup no debería notar el cambio en `src/2`, pero lo hace, debido a las adaptaciones descritas anteriormente.



MODOS



-u, --update

Actualiza recursivamente el índice de las rutas dadas y sus descendientes. Se deben especificar una o más rutas, y si una ruta termina con un enlace

simbólico, se indexará el enlace en sí, no el destino. Si no se especifica ninguna opción de modo,

La actualización es la opción predeterminada y las rutas pueden excluirse mediante las opciones `exclude`, `-excluderx` y `onfilesystem`.

innecesarias de archivos que sabes que son aburridos.

> `--fake-invalid`

marcar rutas específicas como no actualizadas, forzando la próxima ejecución de “bup save” a volver a verificar su contenido.

> `-f, --indexfile=indexfile`

utilice un nombre de archivo de índice diferente en lugar de \$BUP_DIR/bupindex.

> `exclude=ruta`

excluir ruta de la copia de seguridad (puede repetirse).

> `excludefrom=nombre de archivo`

Lectura: excluye rutas del nombre de archivo, una ruta por línea (puede repetirse). Ignora las líneas completamente vacías.

> `excluderx=patrón`

excluye cualquier patrón de coincidencia de ruta, que debe ser una expresión regular de Python (<http://docs.python.org/library/re.html>).

El patrón se comparará con la ruta completa, sin anclaje, por lo que «x/y» coincidirá con «ox/yard» o «box/yards». Para excluir el contenido de /tmp, pero no el directorio en sí, use «^/tmp/.*». (puede repetirse)

EJEMPLOS:

<code>`/foo\$' -</code>	excluye cualquier archivo llamado foo
<code>`/foo/\$' -</code>	excluye cualquier directorio llamado foo
<code>`/foo/.*' -</code>	excluye el contenido de cualquier directorio llamado foo
<code>`^/tmp/.*' -</code>	Excluye el contenido de /tmp a nivel raíz, pero no el propio /tmp.

> `exclude-rx-from=filename`

excluderx patrones de filename, un patrón por línea (puede repetirse). Ignora las líneas completamente vacías.

> `--no-check-device`

no marque una entrada como inválida si el número de dispositivo (stat(2) st_dev) cambia. Esto puede ser útil al indexar sistemas de archivos remotos, montados automáticamente o de instantáneas (LVM, Btrfs, etc.), donde el número de dispositivo no es fijo. v,

> `-v, --verbose`

aumenta la salida del registro durante la actualización (puede usarse más de una vez). Con un v, se imprime cada directorio a medida que se actualiza; con dos v, se imprime también cada archivo.



EJEMPLOS

```
bup index -vux /etc/var/usr
```



VER TAMBIÉN

Bup-Save →

bup-drecurse →

Bup-On →



BUP

Part of the *Bup* → suite.



EN INGLÉS - ORIGINAL

This ensures that no subsequent changes to those paths can result in timestamps that are identical to those in the index. If that were possible, bup could overlook the modifications.

You can see the effect of this behavior in this example (assume that less than one second elapses between the initial file creation and first index run):

```
$ touch src/1 src/2
# A "sleep 1" here would avoid the unexpected save.
$ bup index src
$ bup save -n src src # Saves 1 and 2.
$ date > src/1
$ bup index src
$ date > src/2      # Not indexed.
$ bup save -n src src # But src/2 is saved anyway.
```

Strictly speaking, bup should not notice the change to src/2, but it does, due to the accommodations described above.

MODES

-u, --update

recursively update the index for the given paths and their descendants. One or more paths must be specified, and if a path ends with a symbolic link, the link itself will be indexed, not the target. If no mode option is given, --update is the default, and paths may be excluded by the --exclude, --exclude-rx, and --one-file-system options.

-p, --print

print the contents of the index. If paths are given, shows the given entries and their descendants. If no paths are given, shows the entries starting at the current working directory (.).

-m, --modified

prints only files which are marked as modified (ie. changed since the most recent backup) in the index. Implies -p.

-s, --status

prepend a status code (A, M, D, or space) before each path. Implies -p. The codes mean, respectively, that a file is marked in the index as added, modified, deleted, or unchanged since the last backup.



--check

carefully check index file integrity before and after updating. Mostly useful for automated tests.

--clear

clear the default index.

OPTIONS

-H, --hash

for each file printed, prepend the most recently recorded hash code. The hash code is normally generated by bup save. For objects which have not yet been backed up, the hash code will be 00000000000000000000000000000000. Note that the hash code is printed even if the file is known to be modified or deleted in the index (ie. the file on the filesystem no longer matches the recorded hash). If this is a problem for you, use --status.

-l, --long

print more information about each file, in a similar format to the -l option to [ls\(1\)](#).

-x, --xdev, --one-file-system

don't cross filesystem boundaries when traversing the filesystem – though as with tar and rsync, the mount points themselves will still be indexed. Only applicable if you're using -u.

--fake-valid

mark specified paths as up-to-date even if they aren't. This can be useful for testing, or to avoid unnecessarily backing up files that you know are boring.

--fake-invalid

mark specified paths as not up-to-date, forcing the next “bup save” run to re-check their contents.

-f, --indexfile=*indexfile*

use a different index filename instead of \$BUP_DIR/bupindex.

--exclude=*path*

exclude *path* from the backup (may be repeated).

--exclude-from=*filename*

read –exclude paths from *filename*, one path per-line (may be repeated). Ignore completely empty lines.

--exclude-rx=*pattern*

exclude any path matching *pattern*, which must be a Python regular expression (<http://docs.python.org/library/re.html>). The pattern will be compared against the full path, without anchoring, so “x/y” will match “ox/yard” or “box/yards”. To exclude the contents of /tmp, but not the directory itself, use “^/tmp/.”. (may be repeated)

Examples:

- `'/foo$'` - exclude any file named foo
- `'/foo/$'` - exclude any directory named foo
- `'/foo/.'` - exclude the content of any directory named foo
- `'^/tmp/.'` - exclude root-level /tmp's content, but not /tmp itself

--exclude-rx-from=*filename*

read –exclude-rx patterns from *filename*, one pattern per-line (may be repeated). Ignore completely empty lines.

--no-check-device

don't mark an entry invalid if the device number ([stat\(2\)](#) st_dev) changes. This can be useful when indexing remote, automounted, or snapshot filesystems (LVM, Btrfs, etc.), where the device number isn't fixed.

-v, --verbose

increase log output during update (can be used more than once). With one -v, print each directory as it is updated; with two -v, print each file too