

NAME

shred – overwrite a file to hide its contents, and optionally delete it

SYNOPSIS

shred [*OPTION*]... *FILE*...

DESCRIPTION

Overwrite the specified FILE(s) repeatedly, in order to make it harder for even very expensive hardware probing to recover the data.

If FILE is –, shred standard output.

Mandatory arguments to long options are mandatory for short options too.

-f, --force

change permissions to allow writing if necessary

-n, --iterations=*N*

overwrite *N* times instead of the default (3)

--random-source=*FILE*

get random bytes from FILE

-s, --size=*N*

shred this many bytes (suffixes like K, M, G accepted)

-u

deallocate and remove file after overwriting

--remove[=*HOW*]

like **-u** but give control on HOW to delete; See below

-v, --verbose

show progress

-x, --exact

do not round file sizes up to the next full block;

this is the default for non-regular files

-z, --zero

add a final overwrite with zeros to hide shredding

--help display this help and exit

--version

output version information and exit

Delete FILE(s) if **--remove** (**-u**) is specified. The default is not to remove the files because it is common to operate on device files like */dev/hda*, and those files usually should not be removed. The optional HOW parameter indicates how to remove a directory entry: 'unlink' => use a standard unlink call. 'wipe' => also first obfuscate bytes in the name. 'wipesync' => also sync each obfuscated byte to disk. The default mode is 'wipesync', but note it can be expensive.

CAUTION: Note that shred relies on a very important assumption: that the file system overwrites data in place. This is the traditional way to do things, but many modern file system designs do not satisfy this assumption. The following are examples of file systems on which shred is not effective, or is not guaranteed to be effective in all file system modes:

- * log-structured or journaled file systems, such as those supplied with AIX and Solaris (and JFS, ReiserFS, XFS, Ext3, etc.)

- * file systems that write redundant data and carry on even if some writes fail, such as RAID-based file systems

- * file systems that make snapshots, such as Network Appliance's NFS server

- * file systems that cache in temporary locations, such as NFS version 3 clients

* compressed file systems

In the case of ext3 file systems, the above disclaimer applies (and shred is thus of limited effectiveness) only in data=journal mode, which journals file data in addition to just metadata. In both the data=ordered (default) and data=writeback modes, shred works as usual. Ext3 journaling modes can be changed by adding the data=something option to the mount options for a particular file system in the */etc/fstab* file, as documented in the mount man page (man mount).

In addition, file system backups and remote mirrors may contain copies of the file that cannot be removed, and that will allow a shredded file to be recovered later.

AUTHOR

Written by Colin Plumb.

REPORTING BUGS

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>

Report shred translation bugs to <<https://translationproject.org/team/>>

COPYRIGHT

Copyright © 2018 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

Full documentation at: <<https://www.gnu.org/software/coreutils/shred>>
or available locally via: info '(coreutils) shred invocation'