NAME

remsync - remotely synchronize file trees

SYNOPSIS

remsync −sxv *tree* [*state-file*]
remsync −duxvD *tree* [*state-file* [*diff-file*]]
remsync [**−xv**] *tree* [*diff-file*]

DESCRIPTION

Remsync synchronizes file trees of distant machines, i.e. machines that do not have a fast network between them. It accomplishes this in three steps:

Create a state file containing a description of the machine to be updated.

Compute a file of differences on the source machine using the state file to compare the two file trees.

Update the target machine using the data in the differences file.

This process requires that you move two files, a state file from the target machine to the source machine, and a differences file from the source machine to the target machine. The state file is an ASCII file that may be edited, usually to make **remsync** ignore some files or file trees.

The argument *tree* may be a single file or a directory. A directory is traversed recursively. The *state-file* and *diff-file* arguments may be of any file type. The differences file contains an end marker, so it may be followed by trailing junk. Standard input or output is used if these arguments are omitted or replaced by a minus sign.

State file format

A state file has a line for each file in a tree. A line looks like this formally for a simple file:

name mode owner group length date [*link-number* [**last**]]

The best way to show how each type of file is represented is by example:

/	d755	0	0			
bin	d755	2	0			
[	644	2	0	233	759160857	1
cat	755	2	0	3772	768742021	
test	755	2	0	233	759160857	1
dev	d755	0	0			
fd0	b666	0	0	200		
console	c600	10	0	400		
sd2	b600	0	0	a02		
fifo	p700	2	0			
opt	->	usr/opt				
usr	ignore (Cross-device link)					

The root of the tree is always represented by a /, no matter what type of file it may be. Directory entries of the root follow at the same level. Files in subdirectories are indented by two spaces. (Eight spaces are replaced by a TAB.) Normal files have their length and modified time in the state file, devices have their device number in hex, etc. If files are hard linked to each other then they all get an extra "link number" to bind them together. The last link is marked with the word **last**.

One usually only modifies a state file to ignore differences between two files. One does this by replacing the file attributes with the word **ignore**. (**Remsync** generates this keyword too, with the reason why added in parentheses.)

OPTIONS

−s Generate a state file.

−d Generate a differences file. (The default is to apply a differences file.)

−u Only add new files or update files with newer versions.

−x Do not cross device boundaries. This allows one to operate on the root file system for instance ignoring the **/usr** file system.

−D Debug differences file generation. With this flag no file contents are added to the differences file. The result is then human readable.

−v Lists the commands added to the differences file, or the actions done applying a differences file. The output looks like UNIX commands except for the words "add", "restore" and "update" indicating addition of a new file, replacing a file with an older version, or replacement by a newer version.

EXAMPLES

Actions taken by the author to update his notebook "finiah" from his main machine "darask":

finiah# remsync -s /usr /tmp/finiah.state

Edit the state file to ignore .Xauthority files and /usr/var.

finiah# tar cvf /dev/fd0 /tmp/finiah.state

darask# tar xvf /dev/fd0
darask# remsync -dv /usr /tmp/finiah.state | vol 1440 /dev/fd0

finiah# vol 1440 /dev/fd0 | remsync -v /usr

One could add a file compression/decompression program between **remsync** and **vol**, to reduce the number of floppies to move about, but that actually slows things down! (Note that one only needs to shuffle two floppies about if the two machines are adjacent. To update a remote machine it does make sense to use compression to reduce the number of floppies to carry.)

SEE ALSO

synctree(1), **vol**(1), **tar**(1).

NOTES

Nothing stops you from using **remsync** over a fast network of course. **Synctree** can be a bit tedious if you only want to ignore a few files. Editing a state file is then easier.

BUGS

Files are overwritten, not removed, when they are updated. This means that links outside the tree are also updated. The less desirable alternative to this is to break the link before the update.

The verbose option may say that a link is to be created when making a differences file. The link is often already there when the update takes place, so no action is taken, and thus no talk about it. So you may miss a few mutterings about links if you compare the messages.

AUTHOR

Kees J. Bot (kjb@cs.vu.nl)