

chown simply means Change Owner

Changes the owner or group associated with a file, directory, group or specific symbolic link

Allows you to separate access to files, by assigning multiple groups to users you can define specifically who and how they can access certain files.

symbolic link - is a special type of file that contains a reference to another file or directory
fast symlink is stored in data structure
slow symlink is stored in text file

- f Don't report any failure to change file owner or group, nor modify the exit status to reflect such failures.
- H If the -R option is specified, symbolic links on the command line are followed. (Symbolic links encountered in the tree traversal are not followed.)
- h If the file is a symbolic link, change the user ID and/or the group ID of the link itself.
- L If the -R option is specified, all symbolic links are followed.
- P If the -R option is specified, no symbolic links are followed. Instead, the user and/or group ID of the link itself are modified. This is the default. Use -h to change the user ID and/or the group of symbolic links.
- R Change the user ID and/or the group ID for the file hierarchies rooted in the files instead of just the files themselves.

Examples will only edit group on directories and files as I have only one user/owner to choose from

chmod used to edit permissions

- f (--silent, --quiet) suppresses most usage/error messages
- v (--verbose) returns a diagnostic of all files processed
- R (--recursive) will change owner or group on everything recursively in the directory unless a specific file or directory is include in the specified path
- h (--no-dereference) changes only the symbolic link (useful only on systems that allow changes to symbolic links)
- P do not traverse any symbolic links
- L traverse every symbolic encountered
- H if a command line argument is a symbolic link to a directory, traverse it

chown -R ben2d2:staff ben_test

chown -h ben2d2:staff ben_test

chown -P ben2d2:staff ben_test/file1.txt (changes

chown -P -R -v ben2d2:_developer ben_test/pics/. (changes all files and directory)

chown -v -R ben2d2: [GROUP] ben_test

Some examples of how the owner/group can be specified:

OWNER

If only an OWNER (a user name or numeric user id) is given, that user is made the owner of each given file, and the files' group is not changed.

OWNER.GROUP

OWNER:GROUP

If the OWNER is followed by a colon or dot and a GROUP (a group name or numeric group id), with no spaces between them, the group ownership of the files is changed as well (to GROUP).

OWNER.

OWNER:

If a colon or dot but no group name follows OWNER, that user is made the owner of the files and the group of the files is changed to OWNER's login group.

.GROUP

:GROUP

If the colon or dot and following GROUP are given, but the owner is omitted, only the group of the files is changed; in this case, 'chown' performs the same function as 'chgrp'.

Resources:

man pages for chown

<http://linux.die.net/man/1/chown>

<http://aplawrence.com/Unixart/chown.html>

<http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds1/chown.htm>