NAME

chown, **chgrp** — change file ownership

SYNOPSIS

chown [**-h**]|[**-R** [**-P**|**-H**|**-L**] [**-x**]] [**-fvc**] [**--**[**no-**]**preserve-root**]
  [**-F** *from-user*[**:**[*from-group*]]|**:***from-group*] *user*[**:**[*group*]]|**:***group* *file*...
chown [**-h**]|[**-R** [**-P**|**-H**|**-L**] [**-x**]] [**-fvc**] [**--**[**no-**]**preserve-root**]
  [**-F** *from-user*[**:**[*from-group*]]|**:***from-group*] **-r** *ref-file* *file*...

chgrp [**-h**]|[**-R** [**-P**|**-H**|**-L**] [**-x**]] [**-fvc**] [**--**[**no-**]**preserve-root**]
  [**-F** *from-user*[**:**[*from-group*]]|**:***from-group*] *group* *file*...
chgrp [**-h**]|[**-R** [**-P**|**-H**|**-L**] [**-x**]] [**-fvc**] [**--**[**no-**]**preserve-root**]
  [**-F** *from-user*[**:**[*from-group*]]|**:***from-group*] **-r** *ref-file* *file*...

DESCRIPTION

Changes the owner and/or group of *file*s to match the specification or *ref-file*'s; with **-F** — only if the *file* already had that owner and/or group.

Changing the group can be done by the owner to any of the user's current groups. Changing the owner is always privileged. For non-directories, changing ownership clears: the set-user-ID permission bit, any capabilities (under Linux), the set-group-ID permission bit (if group-executable).

chown and -F formats

username, *UID*	That user; no group.
username:	That user and the corresponding login group.
{*username*, *UID*}:{*groupname*, *GID*}	That user and group.
:*group*, :*GID*	No user; that group.

In all cases, a user or group name takes precedence over the ID (i.e. if a **1312** user exists, that argument will resolve thereto, rather than UID **1312**).

OPTIONS

-h, **--no-dereference** Do not follow symbolic link *file*s (without **-R**), or *ref-file*.

---dereference Do. This is the default.

-R, **--recursive** Change ownership of all of *file*s' descendants, as well.

-P Don't follow any symbolic links during the descent. This is the default.

-H Only follow *file*s, but not any of their descendants. This is the moral equivalent of **-h**.

-L Follow all symbolic links.

-x, **--one-file-system** Don't change ownership of or descend to filesystems (mount-points) different than their corresponding *file*.

-f, **--quiet**, **--silent** Don't write stat(2) and chown(2) errors to the standard error stream.

-v, **--verbose** Log all processed files to the standard output stream.

-c, **--changes** Log only files whose ownership was different than what it was changed to.

--no-preserve-root Allow files equivalent to /. This is the default.

--preserve-root Refuse to process these files.

-F, **--from**=*from-user*[**:**[*from-group*]]|**:***from-group*
Only change ownership of files already owned by *from-user* and/or *from-group*.

-r, **--reference**=*ref-file* Use ownership of *ref-file*. **chgrp** only uses the group ownership.

EXIT STATUS

1 if *ref-file*, *file*, or one of its descendants with **-RL** didn't exist, a *file* (or its descendant with **-RL**) was / and **--preserve-root** was specified, or ownership could not be changed.

EXAMPLES

Set the group of default-created files in images/ to qemu:

```
$ chown -RvF cicada: :qemu images/
chown: images/potoooooooos: not cicada:users: ignoring fried:qemu
chown: images/klepis: users -> qemu
chown: images/shadowfax: not cicada:users: ignoring cicada:adm
chown: images/: not cicada:users: ignoring qemu:qemu
```

SEE ALSO

chmod(1), chown(2), passwd(5)

STANDARDS

Conforms to IEEE Std 1003.1-2024 ("POSIX.1"); **-hRPHL** are the only flags specified by the standard, with exclusive **-hR**. The only standard-mandated **chown** format is *user*[**:***group*].

-fvc, **--preserve-root**, **--no-preserve-root**, **--reference**, and **--from** are extensions, compatible from the GNU system. **-x**, short **-rF**, and **--from** in **chgrp** are extensions.

-fv are also present in FreeBSD and NetBSD. **-x** is also present in FreeBSD 9.0. **--reference** is also present in NetBSD 7.0.

HISTORY

Research UNIX

chown appears in the first edition of the UNIX Programmer's Manual as chown(I):

```
    NAME        chown  --  change owner
    SYNOPSIS    chown owner file$_1$ ...
```

With the owner, in addition to root, being allowed to change the ownership, but only if the set-user-ID bit was unset, since, as a kernel comment notes, it "could create Trojan Horses".

The owner may be decimal, which takes precedent (if the first character of *owner* is a digit), or a name as it appeared in /etc/uids (the world-readable companion to /etc/passwd, which "should be updated along with the password file").

Version 3 AT&T UNIX sees encrypt(III)ed passwords in /etc/passwd, hence its world-readability, and, hence, use as the single point of ID look-up. chown(II) unconditionally clears the set-user-ID bit instead of denying the operation if it's set.

Version 4 AT&T UNIX preserves it if done by root or real ID of the new owner — i.e. if **chown** is invoked by the new owner (possibly transitively) set-user-ID current owner.

Version 5 AT&T UNIX only persists the set-user-ID bit for root.

Version 6 AT&T UNIX says it forbids ownership changes to anyone but root "in order to simplify as yet unimplemented accounting procedures", and, hence, moves the manual to section VIII. But doesn't actually do it, and the owner can still give away files. The set-user-ID is no longer cleared, either. chgrp(VIII) appears for the first time as well, with identical restrictions and manual page, drawing upon /etc/group.

Version 7 AT&T UNIX keeps up the lie and merges the manuals into a common chown(1).

The BSD

3BSD sees an off-by-default (and otherwise undocumented) **-D**ERNIE kernel macro, controlling clearing of set-user- and -group-ID bits. 4BSD moves the page back to section 8, executables to /etc, and replaces previous macros with a strictly controlled and documented list, among which INSECURE, off-by-default, enables "old style non-vanishing suid bits" — in a default configuration, both bits are cleared.

4.2BSD sees

chgrp [-f] group file ...
/etc/chown [**–f**] owner file ...

With **-f** silencing any errors, **chgrp** moved back to **/bin** and chgrp(1), and restrictions relaxed to allow **chgrp**ing by the owner, so long as they also belong to the target group. However, this is entirely artificial, and chown(8)'s "Only the super-user can change owner, in order to simplify accounting procedures." stanza is a documentation error: both operations (chown(2)) are permitted for root and the owner, but the set-user- and -group-ID bits aren't cleared for root, like in Version 6 AT&T UNIX.

4.3BSD sees

chgrp [-f -R] group file ...
/etc/chown [**–f –R**] owner[.group] file ...

4.3BSD–Tahoe **chown** supports, though undocumented, *.group* and *owner.* forms, both setting only the specified arguments.

4.3BSD–Reno moves **chown** to the familiar /usr/sbin, with the appropriate **SYNOPSIS** change, and **chgrp** to /usr/bin.

Changing the owner is only allowed for root, and changing the group is also allowed if the caller is a member — as present-day. If the owner (group) was changed by not root, the appropriate set-user- or -group-ID bit is cleared.

4.4BSD sees

chown [**-HRfh**] *owner* [*:group*] *file ...*
chown [**-HRfh**] *:group file ...*
chgrp [**-HRfh**] *group files ...*

With **-H** as present-day, and **-h** like present-day **-L**, though both independent of **-R**.

User and group names are given precedence over IDs, for compatibility with IEEE Std 1003.2 ("POSIX.2"). Symlink ownership is unchangeable as a system limitation, making these operations effectively successful no-ops thereon.

4.4BSD–Lite2 Sees [**-HRfh**] replaced with [**-R** [**-H -| -L -| -P**]] [**-f**], as present-day. Symbolic links have lost owners entirely.

System V

Sees Version 7 AT&T UNIX **chown** and **chgrp**, except with set-user- and -group-ID clearing for non-root, until AT&T System V Release 3 UNIX, which validates numeric IDs to be *ushort*s (16-bit), rather than blindly passing through *int*s from atoi(3) — 16 bits on the PDP-11, but 32 bits on the 3B2.

AT&T System V Release 4 UNIX sees

```
chown [-h] [-R] uid file ...
chgrp [-h] [-R] gid file ...
```

with **-R** defaulting to **-L** mode and **-h** acting like present-day **-h** *and* **-P** mode.

A recurring comment in the filesystems' ***setattr**() functions notes:

```
/*
 * Change file ownership; must be the owner of the file
 * or the super-user.  If the system was configured with
 * the "rstchown" option, the owner is not permitted to
 * give away the file, and can change the group id only
 * to a group of which he or she is a member.
 */
```

Indeed, the RSTCHOWN tunable enables a kernel-wide mode equivalent to present-day behaviour (_POSIX_CHOWN_RESTRICTED). The set-user-ID bit is always cleared for non-root, and the set-group-ID bit is cleared if the file is group-executable — this can be thought of as the budding origin of the modern behaviour.

Standards

X/Open Portability Guide Issue 2 (“XPG2”) describes **chown** and **chgrp** as they’re found in AT&T UNIX, permitting any sort of restriction, but outlining that "some systems" permit ⟨root and owner⟩, ⟨just root⟩, or ⟨root and owner (if in groups)⟩ limitations. The name/ID precedence is unspecified except for hinted at with "The owner may be either a decimal user ID or a login name found in the User Database", and likewise for the group.

IEEE Std 1003.2-1992 (“POSIX.2”) sees

```
chown [-R] owner[:group] file ...
chown [-R] group file ...
```

and specifies an explicit name-over-numerical-ID precedent. Symbolic links are always followed as-specified, but

> For implementations on which symbolic links are supported, actual use of the *chown*() function to implement this utility might not be the appropriate, depending on the implementation.

The **:** separator is a refinement of the 4.3BSD **.** syntax, since users and groups are allowed to have names that contain "**.**"s, but no systems support ones with "**:**"s.

IEEE Std 1003.1-2001 (“POSIX.1”) adds **-h** and **-PHL**, as present-day.