

NAME

dotlockfile – Utility to manage lockfiles

SYNOPSIS

```
dotlockfile -l [-r retries] [-i interval] [-p] [-q] <-m | lockfile>
dotlockfile -l [-r retries] [-i interval] [-p] [-q] <-m | lockfile> [-P] cmd args ...
dotlockfile -u | -t
```

DESCRIPTION

dotlockfile is a command line utility to reliably create, test and remove lockfiles. It creates lockfiles *reliably* on local and NFS filesystems, because the crucial steps of testing for a preexisting lockfile and creating it are performed *atomically* by a *single* call to *link*(2). *Manpageloc kfile_create*(3) describes the used algorithm.

dotlockfile is installed with attribute **SETGID** *mail* and thus can also be used to lock and unlock mailboxes *even* if the mailspool directory is only writable by group mail.

The name **dotlockfile** comes from the way mailboxes are locked for updates on a lot of UNIX systems. A lockfile is created with the same filename as the mailbox but with the string ".lock" appended.

The names **dotlock** and **lockfile** were already taken – hence the name **dotlockfile** :).

OPTIONS

-l Create a lockfile if no preexisting valid lockfile is found, else wait and retry according to option **-r**. Retry interval can be explicitly set with option **-i**. This option (**-l**) is the default, so it can be left off.

A lockfile is treated as valid,

- if it holds the *process-id* of a running process,
- or if it does not hold any *process-id* and has been touched less than 5 minutes ago (timestamp is younger than 5 minutes).

-r retries

The number of times **dotlockfile** retries to acquire the lock if it failed the first time before giving up. The initial sleep after failing to acquire the lock is 5 seconds. After each retry the sleep interval is increased incrementally by 5 seconds up to a maximum sleep of 60 seconds between tries unless overridden by **-i**. The default number of retries is 5. To try only once, use "**-r 0**". To try indefinitely, use "**-r -1**".

-i interval

Sets a consistent retry interval.

-u Remove a lockfile.

-t Touch an existing lockfile (update the timestamp). Useful for lockfiles on NFS filesystems. For lockfiles on local filesystems the **-p** option is preferable.

-p Write the *process-id* of the calling process (or dotlockfile itself if a command is executed) into the lockfile. Also when testing for an existing lockfile, check the contents for the *process-id* of a running process to verify if the lockfile is still valid. Obviously useful only for lockfiles on local filesystems.

-m Lock or unlock the current users mailbox. The path to the mailbox is the default system mailspool directory (usually */var/mail*) with the username as gotten from *getpwuid*() appended. If the environment variable *\$MAIL* is set, that is used instead. Then the string ".lock" is appended to get the name of the actual lockfile.

-q Don't print warnings or errors to the standard error output. Used internally by liblockfile when it spawns **dotlockfile** as a helper program.

-P On successful "lock and spawn command", don't exit with status zero, but pass through the exit value of the spawned command.

lockfile The lockfile to be created or removed. Must not be specified if the **-m** option is given.

command argument ...

Create lockfile, run the *command*, wait for it to exit, and remove lockfile.

RETURN VALUE

Zero on success, and non-zero on failure. When locking (the default, or with the **-l** option) **dotlockfile** returns the same values as the library function *lockfile_create(3)*. Unlocking a non-existent lockfile is not an error.

Unless the **-P** option was supplied, when a command is executed, the return value does not correspond with that of the command that was run. If locking and unlocking was successful, the exit status is zero.

NOTES

The lockfile is created exactly as named on the command line. The extension *".lock"* is *not* automatically appended.

This utility is a lot like the *lockfile(1)* utility included with *procmail*, and the *mutt_dotlock(1)* utility included with *mutt*. However the command-line arguments differ, and so does the return status. It is believed, that *dotlockfile* is the most flexible implementation, since it automatically detects when it needs to use privileges to lock a mailbox, and does it safely.

The above mentioned *lockfile_create(3)* manpage is present in the *liblockfile-dev* package.

BUGS

None known.

SEE ALSO

lockfile_create(3), *maillock(3)*

AUTHOR

Miquel van Smoorenburg