daemonize

NAME

daemonize − run a program as a Unix daemon

SYNOPSIS

daemonize [-a] [-c *directory*] [-e *stderr*] [-o *stdout*] [-p *pidfile*] [-l *lockfile*] [-u *user*] [-v] *path [arg]* ...

DESCRIPTION

daemonize runs a command as a Unix daemon. As defined in W. Richard Stevens' 1990 book, *Unix Network Programming* (Addison-Wesley, 1990), a daemon is "a process that executes 'in the background' (i.e., without an associated terminal or login shell) either waiting for some event to occur, or waiting to perform some specified task on a periodic basis." Upon startup, a typical daemon program will:

- Close all open file descriptors (especially standard input, standard output and standard error)
- Change its working directory to the root filesystem, to ensure that it doesn't tie up another filesystem and prevent it from being unmounted
- Reset its *umask* value
- Run in the background (i.e., *fork*)

- o Disassociate from its process group (usually a shell), to insulate itself from signals (such as HUP) sent to the process group
- o Ignore all terminal I/O signals
- o Disassociate from the control terminal (and take steps not to reacquire one)
- o Handle any SIGCLD signals

Most programs that are designed to be run as daemons do that work for themselves. However, you'll occasionally run across one that does not. When you must run a daemon program that does not properly make itself into a true Unix daemon, you can use **daemonize** to force it to run as a true daemon.

OPTIONS

-a
: Append to the output files, rather than overwriting them (which is the default). Only applicable if *-e* and/or *-o* are specified.

-c *directory*
: Specifies the directory to which to change before running the program. Defaults to "/". The choice for this option is important. The file system containing the daemon's working directory cannot be unmounted while the daemon is running. That's why most daemons are careful to use a working directory on the root file system.

-e *stderr*
: Redirect standard error to the specified file, instead of "/dev/null".

Warning: Be careful where you redirect the output! The file system containing the open file cannot be unmounted as long as the file is open. For best results, make sure that this output file is on the same file system as the daemon's working directory. (See the *-c* option.)

-E *name=value*
: Add an environment variable to the environment the daemon will see. The parameter must be of the form *name=value*. This parameter may be specified multiple times.

-o *stdout*

Redirect standard output to the specified file, instead of "/dev/null".

Warning: Be careful where you redirect the output! The file system containing the open file cannot be unmounted as long as the file is open. For best results, make sure that this output file is on the same file system as the daemon's working directory. (See the *-c* option.)

-p *pidfile*

Causes **daemonize** to write the numeric process ID (PID) of the running daemon to the specified file. This option is useful when the program being daemonized doesn't create its own PID file.

-l *lockfile*

Single-instance checking. Causes **daemonize** to ensure that no more than one instance of the daemon is running by placing an exclusive lock on given lockfile. If another process already has a lock on the lockfile, *daemonize* exits.

It *is* possible to use the *pidfile* as the lock file (e.g., "-p /var/run/foo -l /var/run/foo"), though typical daemons use separate files.

NOTE: If the executed program decides to close all file descriptors, the single-instance locking will not work, since the lock depends on an open file descriptor. (The operating system kernel removes the lock once the process holding the lock closes the file or exits.) Normal processes that do not daemonize themselves do not usually close all file descriptors.

-u *user*

Run the program as the specified user. This option only works if **daemonize** is invoked by the superuser. Note: For obvious reasons, it's very dangerous to install **daemonize** as a *setuid-to-root* executable. For that reason, **daemonize** will refuse to run if it detects that it has been installed that way.

-v

Cause **daemonize** to write verbose messages to standard error, telling what it's doing as it daemonizes the program.

NOTES

If the host operating system provides the **daemon**(3) library routine, **daemonize** will use it. Otherwise, **daemonize** uses its own version of

daemon(3). This choice is made at compile time. (BSD 4.4-derived operating systems tend to provide their own **daemon**(3) routine.)

FreeBSD 5.0 introduced a **daemon**(1) command that is similar to, but less functional, than **daemonize**.

LICENSE

This program released under a BSD-style license. For more details, consult the LICENSE file accompanying the source distribution, or visit "http://software.clapper.org/daemonize/LICENSE".

SEE ALSO

daemon(3), **setsid**(2), **flock(2)**

daemonize Home Page: http://software.clapper.org/daemonize/

AUTHOR

Brian M. Clapper, *bmc <at> clapper <dot> org*

CONTRIBUTORS

Support for the *-e* and *-o* options is based on a patch submitted by Tim Starling *(tstarling <at> wikimedia <dot> org)*.

Support for the *-l* option is based on a patch submitted by Yakov Lerner *(iler.ml <at> gmail <dot> com)*.