NAME

killall - kill processes by name

SYNOPSIS

killall [**-delmsvz**] [**-help**] [**-I**] [**-j** *jail*] [**-u** *user*] [**-t** *tty*] [**-c** *procname*] [-*SIGNAL*] [*procname* ...]

DESCRIPTION

The **killall** utility kills processes selected by name, as opposed to the selection by PID as done by kill(1). By default, it will send a TERM signal to all processes with a real UID identical to the caller of **killall** that match the name *procname*. The super-user is allowed to kill any process.

The options are as follows:

-d Be more verbose about what will be done, but do not send any signal. The total number of user processes and the real user ID is shown. A list of the processes that will be sent the signal will be printed, or a message indicating that no matching processes have been found.

-e Use the effective user ID instead of the (default) real user ID for matching processes specified with the **-u** option.

-help Give a help on the command usage and exit.

-I Request confirmation before attempting to signal each process.

-l List the names of the available signals and exit, like in kill(1).

-m Match the argument *procname* as a (case sensitive) regular expression against the names of processes found. CAUTION! This is dangerous, a single dot will match any process running under the real UID of the caller.

-v Be verbose about what will be done.

-s Same as **-v**, but do not send any signal.

-*SIGNAL* Send a different signal instead of the default TERM. The signal may be specified either as a name (with or without a leading "SIG"), or numerically.

-j *jail* Kill processes in the specified *jail*.

-u *user* Limit potentially matching processes to those belonging to the specified *user*.

-t *tty* Limit potentially matching processes to those running on the specified *tty*.

-c *procname* Limit potentially matching processes to those matching the specified *procname*.

-q Suppress error message if no processes are matched.

-z Do not skip zombies. This should not have any effect except to print a few error messages if there are zombie processes that match the specified pattern.

ALL PROCESSES

Sending a signal to all processes with the given UID is already supported by kill(1). So use kill(1) for this job (e.g. "kill -TERM -1" or as root "echo kill -TERM -1 | su -m <user>").

IMPLEMENTATION NOTES

This FreeBSD implementation of **killall** has completely different semantics as compared to the traditional UNIX System V behavior of **killall**. The latter will kill all processes that the current user is able to kill, and is intended to be used by the system shutdown process only.

EXIT STATUS

The **killall** utility exits 0 if some processes have been found and signalled successfully. Otherwise, a status of 1 will be returned.

EXAMPLES

Send SIGTERM to all firefox processes:

```
killall firefox
```

Send SIGTERM to firefox processes belonging to *USER*:

```
killall -u ${USER} firefox
```

Stop all firefox processes:

```
killall -SIGSTOP firefox
```

Resume firefox processes:

```
killall -SIGCONT firefox
```

Show what would be done to firefox processes, but do not actually signal them:

```
killall -s firefox
```

Send SIGKILL to csh process running inside jail ID 282:

```
killall -9 -j282 csh
```

Send SIGTERM to all processes matching provided pattern (like vim and vimdiff):

```
killall -m 'vim*'
```

DIAGNOSTICS

Diagnostic messages will only be printed if the **-d** flag is used.

SEE ALSO

kill(1), pkill(1), sysctl(3), jail(8)

HISTORY

The **killall** command appeared in FreeBSD 2.1. It has been modeled after the **killall** command as available on other platforms.

AUTHORS

The **killall** program was originally written in Perl and was contributed by Wolfram Schneider, this manual page has been written by Jörg Wunsch. The current version of **killall** was rewritten in C by Peter Wemm using sysctl(3).