NAME

pgrep, **pkill** - find or signal processes by name

SYNOPSIS

pgrep [**-LSafilnoqvx**] [**-F** *pidfile*] [**-G** *gid*] [**-M** *core*] [**-N** *system*] [**-P** *ppid*] [**-U** *uid*] [**-c** *class*] [**-d** *delim*]
      [**-g** *pgrp*] [**-j** *jail*] [**-s** *sid*] [**-t** *tty*] [**-u** *euid*] *pattern ...*
pkill [*-signal*] [**-ILafilnovx**] [**-F** *pidfile*] [**-G** *gid*] [**-M** *core*] [**-N** *system*] [**-P** *ppid*] [**-U** *uid*] [**-c** *class*]
      [**-g** *pgrp*] [**-j** *jail*] [**-s** *sid*] [**-t** *tty*] [**-u** *euid*] *pattern ...*

DESCRIPTION

The **pgrep** command searches the process table on the running system and prints the process IDs of all processes that match the criteria given on the command line.

The **pkill** command searches the process table on the running system and signals all processes that match the criteria given on the command line.

The following options are available:

-F *pidfile*   Restrict matches to a process whose PID is stored in the *pidfile* file.

-G *gid*   Restrict matches to processes with a real group ID in the comma-separated list *gid*.

-I   Request confirmation before attempting to signal each process.

-L   The *pidfile* file given for the **-F** option must be locked with the flock(2) syscall or created with pidfile(3).

-M *core*   Extract values associated with the name list from the specified core instead of the currently running system.

-N *system*   Extract the name list from the specified system instead of the default, which is the kernel image the system has booted from.

-P *ppid*   Restrict matches to processes with a parent process ID in the comma-separated list *ppid*.

-S   Search also in system processes (kernel threads).

-U *uid*   Restrict matches to processes with a real user ID in the comma-separated list *uid*.

-d *delim*   Specify a delimiter to be printed between each process ID. The default is a newline. This

option can only be used with the **pgrep** command.

-a   Include process ancestors in the match list. By default, the current **pgrep** or **pkill** process and all of its ancestors are excluded (unless **-v** is used).

-c *class*   Restrict matches to processes running with specified login class *class*.

-f   Match against full argument lists. The default is to match against process names.

-g *pgrp*   Restrict matches to processes with a process group ID in the comma-separated list *pgrp*. The value zero is taken to mean the process group ID of the running **pgrep** or **pkill** command.

-i   Ignore case distinctions in both the process table and the supplied pattern.

-j *jail*   Restrict matches to processes inside the specified jails. The argument *jail* may be "any" to match processes in any jail, "none" to match processes not in jail, or a comma-separated list of jail IDs or names.

-l   Long output. For **pgrep**, print the process name in addition to the process ID for each matching process. If used in conjunction with **-f**, print the process ID and the full argument list for each matching process. For **pkill**, display the kill command used for each process killed.

-n   Select only the newest (most recently started) of the matching processes.

-o   Select only the oldest (least recently started) of the matching processes.

-q   For **pgrep**, Do not write anything to standard output.

-s *sid*   Restrict matches to processes with a session ID in the comma-separated list *sid*. The value zero is taken to mean the session ID of the running **pgrep** or **pkill** command.

-t *tty*   Restrict matches to processes associated with a terminal in the comma-separated list *tty*. Terminal names may be of the form *ttyxx* or the shortened form *xx*. A single dash ('-') matches processes not associated with a terminal.

-u *euid*   Restrict matches to processes with an effective user ID in the comma-separated list *euid*.

-v   Reverse the sense of the matching; display processes that do not match the given criteria.

-x   Require an exact match of the process name, or argument list if **-f** is given. The default is
              to match any substring.

-*signal*   A non-negative decimal number or symbolic signal name specifying the signal to be sent
              instead of the default TERM. This option is valid only when given as the first argument to
              pkill.

If any *pattern* operands are specified, they are used as extended regular expressions to match the
command name or full argument list of each process. If the **-f** option is not specified, then the *pattern*
will attempt to match the command name. However, presently FreeBSD will only keep track of the first
19 characters of the command name for each process. Attempts to match any characters after the first 19
of a command name will quietly fail.

Note that a running **pgrep** or **pkill** process will never consider itself nor system processes (kernel
threads) as a potential match.

IMPLEMENTATION NOTES
The Sun Solaris implementation utilised procfs to obtain process information. This implementation
utilises kvm(3) instead. On a live system, kvm(3) uses *kern.proc* MIB to obtain the list of processes,
kernel memory through */dev/kmem* is not accessed.

EXIT STATUS
The **pgrep** and **pkill** utilities return one of the following values upon exit:

0   One or more processes were matched.

1   No processes were matched.

2   Invalid options were specified on the command line.

3   An internal error occurred.

EXAMPLES
Show the pid of the process holding the */tmp/.X0-lock* pid file:

    $ pgrep -F /tmp/.X0-lock
    1211

Show the pid and the name of the process including kernel threads in the search:

```
$ pgrep -lS vnlru
37 vnlru
```

Search for processes including kernel threads that match the extended regular expression pattern:

```
$ pgrep -S 'crypto.*[2-3]'
20
19
6
5
```

Show long output for firefox processes:

```
$ pgrep -l firefox
1312 firefox
1309 firefox
1288 firefox
1280 firefox
1279 firefox
1278 firefox
1277 firefox
1264 firefox
```

Same as above but just showing the pid of the most recent process:

```
$ pgrep -n firefox
1312
```

Look for vim processes. Match against the full argument list:

```
$ pgrep -f vim
44968
30790
```

Same as above but matching against the 'list' word and showing the full argument list:

```
$ pgrep -f -l list
30790 vim list.txt
```

Send *SIGSTOP* signal to processes that are an exact match:

    $ pkill -SIGSTOP -f -x "vim list.txt"

  Without **-f** names over 19 characters will silently fail:

    $ vim this_is_a_very_long_file_name &
    [1] 36689
    $

    [1]+ Stopped   vim this_is_a_very_long_file_name
    $ pgrep "vim this"
    $

  Same as above using the **-f** flag:

    $ pgrep -f "vim this"
    36689

  Find the top(1) command running in any jail:

    $ pgrep -j any top
    34498

  Show all processes running in jail ID 58:

    $ pgrep -l -j58 '.*'
    28397 pkg-static
    28396 pkg-static
    28255 sh
    28254 make

COMPATIBILITY
  Historically the option "**-j** 0" means any jail, although in other utilities such as ps(1) jail ID 0 has the
  opposite meaning, not in jail. Therefore "**-j** 0" is deprecated, and its use is discouraged in favor of "**-j**
  any".

SEE ALSO
  kill(1), killall(1), ps(1), flock(2), kill(2), sigaction(2), kvm(3), pidfile(3), re_format(7)

HISTORY
  The **pkill** and **pgrep** utilities first appeared in NetBSD 1.6. They are modelled after utilities of the same

name that appeared in Sun Solaris 7. They made their first appearance in FreeBSD 5.3.

AUTHORS
    Andrew Doran <*ad@NetBSD.org*>