

**NAME**

**pgrep**, **pkill** – look up or signal processes based on name and other attributes

**SYNOPSIS**

**pgrep** [options] pattern

**pkill** [options] pattern

**DESCRIPTION**

**pgrep** looks through the currently running processes and lists the process IDs which match the selection criteria to stdout. All the criteria have to match. For example,

```
$ pgrep -u root sshd
```

will only list the processes called **sshd** AND owned by **root**. On the other hand,

```
$ pgrep -u root,daemon
```

will list the processes owned by **root** OR **daemon**.

**pkill** will send the specified signal (by default **SIGTERM**) to each process instead of listing them on stdout.

**OPTIONS**

**-signal**

**--signal** *signal*

Defines the signal to send to each matched process. Either the numeric or the symbolic signal name can be used. (**pkill** only.)

**-c, --count**

Suppress normal output; instead print a count of matching processes. When count does not match anything, e.g. returns zero, the command will return non-zero value.

**-d, --delimiter** *delimiter*

Sets the string used to delimit each process ID in the output (by default a newline). (**pgrep** only.)

**-f, --full**

The *pattern* is normally only matched against the process name. When **-f** is set, the full command line is used.

**-g, --pgroup** *pgrp*,...

Only match processes in the process group IDs listed. Process group 0 is translated into **pgrep**'s or **pkill**'s own process group.

**-G, --group** *gid*,...

Only match processes whose real group ID is listed. Either the numerical or symbolical value may be used.

**-i, --ignore-case**

Match processes case-insensitively.

**-l, --list-name**

List the process name as well as the process ID. (**pgrep** only.)

**-a, --list-full**

List the full command line as well as the process ID. (**pgrep** only.)

**-n, --newest**

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

**-o, --oldest**

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

**-P, --parent** *ppid*,...

Only match processes whose parent process ID is listed.

- s, --session *sid*,...**  
Only match processes whose process session ID is listed. Session ID 0 is translated into **pgrep**'s or **pgrep**'s own session ID.
- t, --terminal *term*,...**  
Only match processes whose controlling terminal is listed. The terminal name should be specified without the "/dev/" prefix.
- u, --euid *euid*,...**  
Only match processes whose effective user ID is listed. Either the numerical or symbolical value may be used.
- U, --uid *uid*,...**  
Only match processes whose real user ID is listed. Either the numerical or symbolical value may be used.
- v, --inverse**  
Negates the matching. This option is usually used in **pgrep**'s context. In **pgrep**'s context the short option is disabled to avoid accidental usage of the option.
- w, --lightweight**  
Shows all thread ids instead of pids in **pgrep**'s context. In **pgrep**'s context this option is disabled.
- x, --exact**  
Only match processes whose names (or command line if **-f** is specified) **exactly** match the *pattern*.
- F, --pidfile *file***  
Read *PID*'s from file. This option is perhaps more useful for **pgrep** than **pgrep**.
- L, --logpidfile**  
Fail if pidfile (see **-F**) not locked.
- ns *pid***  
Match processes that belong to the same namespaces. Required to run as root to match processes from other users. See **--nslist** for how to limit which namespaces to match.
- nslist *name*,...**  
Match only the provided namespaces. Available namespaces: ipc, mnt, net, pid, user,uts.
- V, --version**  
Display version information and exit.
- h, --help**  
Display help and exit.

## OPERANDS

*pattern* Specifies an Extended Regular Expression for matching against the process names or command lines.

## EXAMPLES

Example 1: Find the process ID of the **named** daemon:

```
$ pgrep -u root named
```

Example 2: Make **syslog** reread its configuration file:

```
$ pkill -HUP syslogd
```

Example 3: Give detailed information on all **xterm** processes:

```
$ ps -fp $(pgrep -d, -x xterm)
```

Example 4: Make all **chrome** processes run nicer:

```
$ renice +4 $(pgrep chrome)
```

**EXIT STATUS**

- |   |  |
|---|--|
| 0 | One or more processes matched the criteria. For <b>pkill</b> the process must also have been successfully signalled. |
| 1 | No processes matched or none of them could be signalled.   |
| 2 | Syntax error in the command line.  |
| 3 | Fatal error: out of memory etc.  |

**NOTES**

The process name used for matching is limited to the 15 characters present in the output of `/proc/pid/stat`. Use the `-f` option to match against the complete command line, `/proc/pid/cmdline`.

The running **pgrep** or **pkill** process will never report itself as a match.

**BUGS**

The options `-n` and `-o` and `-v` can not be combined. Let me know if you need to do this.

Defunct processes are reported.

**SEE ALSO**

**ps**(1), **regex**(7), **signal**(7), **killall**(1), **skill**(1), **kill**(1), **kill**(2)

**AUTHOR**

Kjetil Torgrim Homme <kjetilho@ifi.uio.no>

**REPORTING BUGS**

Please send bug reports to <procps@freelists.org>