NAME

pidof -- find the process ID of a running program.

SYNOPSIS

pidof [**−s**] [**−c**] [**−n**] [**−x**] [**−o** *omitpid[,omitpid...]*] [**−o** *omitpid[,omitpid...]...*] [**−d** *sep*] **program** [**program...**]

DESCRIPTION

Pidof finds the process id's (PIDs) of the named programs. It prints those id's on the standard output. This program is on some systems used in run-level change scripts, especially when the system has a *System-V* like *rc* structure. In that case these scripts are located in /etc/rc?.d, where ? is the runlevel. If the system has a **start-stop-daemon** (8) program that should be used instead.

OPTIONS

−s Single shot - this instructs the program to only return one *pid*.

−c Only return process PIDs that are running with the same root directory. This option is ignored for non-root users, as they will be unable to check the current root directory of processes they do not own.

−n Avoid **stat**(2) system function call on all binaries which are located on network based file systems like **NFS**. Instead of using this option the variable **PIDOF_NETFS** may be set and exported.

−q Do not display matched PIDs to standard out. Simply exit with a status of true or false to indicate whether a matching PID was found.

−x Scripts too - this causes the program to also return process id's of shells running the named scripts.

-d *sep* Tells *pidof* to use *sep* as an output separator if more than one PID is shown. The default separator is a space.

-o *omitpid*

Tells *pidof* to omit processes with that process id. The special pid **%PPID** can be used to name the parent process of the *pidof* program, in other words the calling shell or shell script.

EXIT STATUS

0 At least one program was found with the requested name.

1 No program was found with the requested name.

NOTES

pidof is actually the same program as *killall5*; the program behaves according to the name under which it is called.

When *pidof* is invoked with a full pathname to the program it should find the pid of, it is reasonably safe. Otherwise it is possible that it returns PIDs of running programs that happen to have the same name as the program you're after but are actually other programs. Note that the executable name of running processes is calculated with **readlink**(2), so symbolic links to executables will also match.

Zombie processes or processes in disk sleep (states Z and D, respectively) are ignored, as attempts to access the stats of these will sometimes fail.

SEE ALSO

shutdown(8), **init**(8), **halt**(8), **reboot**(8), **killall5**(8)

AUTHOR

Miquel van Smoorenburg, miquels@cistron.nl