NAME

pstree – display a tree of processes

SYNOPSIS

pstree [–a, −−**arguments**] [–c, −−**compact−not**] [–C, −−**color** *attr*] [–g, −−**show−pgids**] [–h, −−**high-light−all**, –H *pid*, −−**highlight−pid** *pid*] [–l, −−**long**] [–n, −−**numeric−sort**] [–N, −−**ns−sort** *ns*]
[–p, −−**show−pids**] [–s, −−**show−parents**] [–S, −−**ns-changes**] [–t, −−**thread-names**] [–T, −−**hide-threads**] [–u, −−**uid−changes**] [–Z, −−**security-context**] [–A, −−**ascii**, –G, −−**vt100**, –U, −−**unicode**]
[*pid*, *user*]

pstree –V, −−**version**

DESCRIPTION

pstree shows running processes as a tree. The tree is rooted at either *pid* or **init** if *pid* is omitted. If a user name is specified, all process trees rooted at processes owned by that user are shown.

pstree visually merges identical branches by putting them in square brackets and prefixing them with the repetition count, e.g.

```
init−+−getty
     |−getty
     |−getty
     '−getty
```

becomes

```
init−−−4*[getty]
```

Child threads of a process are found under the parent process and are shown with the process name in curly braces, e.g.

```
icecast2−−−13*[{icecast2}]
```

If **pstree** is called as **pstree.x11** then it will prompt the user at the end of the line to press return and will not return until that has happened. This is useful for when **pstree** is run in a xterminal.

Certain kernel or mount parameters, such as the *hidepid* option for procfs, will hide information for some processes. In these situations **pstree** will attempt to build the tree without this information, showing process names as question marks.

OPTIONS

–a   Show command line arguments. If the command line of a process is swapped out, that process is shown in parentheses. **−a** implicitly disables compaction for processes but not threads.

–A   Use ASCII characters to draw the tree.

–c   Disable compaction of identical subtrees. By default, subtrees are compacted whenever possible.

–C   Color the process name by given attribute. Currently **pstree** only accepts the value **age** which col-ors by process age. Processes newer than 60 seconds are green, newer than an hour yellow and the remaining red.

–g   Show PGIDs. Process Group IDs are shown as decimal numbers in parentheses after each process name. **−g** implicitly disables compaction. If both PIDs and PGIDs are displayed then PIDs are shown first.

–G   Use VT100 line drawing characters.

–h   Highlight the current process and its ancestors. This is a no-op if the terminal doesn't support highlighting or if neither the current process nor any of its ancestors are in the subtree being

shown.

–H   Like **–h**, but highlight the specified process instead. Unlike with **–h**, **pstree** fails when using **–H** if highlighting is not available.

–l   Display long lines. By default, lines are truncated to either the COLUMNS environment variable or the display width. If neither of these methods work, the default of 132 columns is used.

–n   Sort processes with the same parent by PID instead of by name. (Numeric sort.)

–N   Show individual trees for each namespace of the type specified. The available types are: *ipc*, *mnt*, *net*, *pid*, *time*, *user*, *uts*. Regular users don't have access to other users' processes information, so the output will be limited.

–p   Show PIDs. PIDs are shown as decimal numbers in parentheses after each process name. **–p** implicitly disables compaction.

–s   Show parent processes of the specified process.

–S   Show namespaces transitions. Like **–N**, the output is limited when running as a regular user.

–t   Show full names for threads when available.

–T   Hide threads and only show processes.

–u   Show uid transitions. Whenever the uid of a process differs from the uid of its parent, the new uid is shown in parentheses after the process name.

–U   Use UTF-8 (Unicode) line drawing characters. Under Linux 1.1-54 and above, UTF-8 mode is entered on the console with **echo –e ' 33%8'** and left with **echo –e ' 33%@'**.

–V   Display version information.

–Z   Show the current security attributes of the process. For SELinux systems this will be the security context.

FILES
/proc   location of the proc file system

BUGS
Some character sets may be incompatible with the VT100 characters.

SEE ALSO
ps(1), **top**(1), **proc**(5).