NAME

ps – display the status of current processes

SYNOPSIS

/usr/ucb/ps [**–acglnrSuUvwx**] [**–t** *term*] [*num*]

AVAILABILITY

SUNWscpu

DESCRIPTION

The **ps** command displays information about processes. Normally, only those processes that are running with your effective user ID and are attached to a controlling terminal (see **termio**(7I)) are shown. Additional categories of processes can be added to the display using various options. In particular, the –**a** option allows you to include processes that are not owned by you (that do not have your user ID), and the –**x** option allows you to include processes without control terminals. When you specify both –**a** and –**x ,** you get processes owned by anyone, with or without a control terminal. The –**r** option restricts the list of processes printed to running and runnable processes.

ps displays in tabular form the process ID, under **PID**; the control terminal (if any), under **TT**; the cpu time used by the process so far, including both user and system time, under **TIME**; the state of the process, under S; and finally, an indication of the **COMMAND** that is running.

The state is given by a single letter from the following:

O	Process is running on a processor.
S	Sleeping. Process is waiting for an event to complete.
R	Runnable. Process is on run queue.
I	Idle. Process is being created.
Z	Zombie state. Process terminated and parent not waiting.
T	Traced. Process stopped by a signal because parent is tracing it.
X	**SXBRK** state. Process is waiting for more primary memory.

OPTIONS

The following options must all be combined to form the first argument:

–**a** Include information about processes owned by others.

–**c** Display the command name, as stored internally in the system for purposes of accounting, rather than the command arguments, which are kept in the process' address space. This is more reliable, if less informative, since the process is free to destroy the latter information.

–**g** Display all processes. Without this option, **ps** only prints interesting processes. Processes are deemed to be uninteresting if they are process group leaders. This normally eliminates top-level command interpreters and processes waiting for users to login on free terminals.

–**l** Display a long listing, with fields **F**, **PPID**, **CP**, **PRI**, **NI**, **SZ**, **RSS** and **WCHAN** as described below.

–**n** Produce numerical output for some fields. In a user listing, the **USER** field is replaced by a **UID** field.

–**r** Restrict output to running and runnable processes.

–S Display accumulated CPU time used by this process and all of its reaped children.

–u Display user-oriented output. This includes fields **USER**, **SZ**, **RSS** and **START** as described below.

–U Update a private database where **ps** keeps system information.

–v Display a version of the output containing virtual memory. This includes fields **SIZE** and **RSS**, described below.

–w Use a wide output format (132 columns rather than 80); if repeated, that is, **–ww ,** use arbitrarily wide output. This information is used to decide how much of long commands to print.

–x Include processes with no controlling terminal.

–t *term* List only process data associated with the terminal, *term*. Terminal identifiers may be specified in one of two forms: the device's file name (for example, **tty04** or **term/14**) or, if the device's file name starts with **tty**, just the digit identifier (for example, **04**).

num A process number may be given, in which case the output is restricted to that process. This option must be supplied last.

DISPLAY FORMATS

Fields that are not common to all output formats:

USER Name of the owner of the process.

NI Process scheduling increment (see **getpriority**(3C) and **nice**(3B)).

SIZE
SZ The combined size of the data and stack segments (in kilobyte units)

RSS Real memory (resident set) size of the process (in kilobyte units).

UID Numerical user-ID of process owner.

PPID Numerical ID of parent of process.

CP Short-term CPU utilization factor (used in scheduling).

PRI The priority of the process (higher numbers mean lower priority).

START The starting time of the process, given in hours, minutes, and seconds. A process begun more than 24 hours before the **ps** inquiry is executed is given in months and days.

WCHAN The address of an event for which the process is sleeping, or in **SXBRK** state (if blank, the process is running).

F Flags (hexadecimal and additive) associated with the process:

- **00** Process has terminated. Process table now available.
- **01** A system process, always in primary memory.
- **02** Parent is tracing process.
- **04** Tracing parent's signal has stopped process. Parent is waiting, see **ptrace**(2).
- **08** Process is currently in primary memory.

10 Process currently in primary memory, locked until an event is completed.

A process that has exited and has a parent, but has not yet been waited for by the parent is marked **< defunct >;** otherwise, **ps** tries to determine the command name and arguments given when the process was created by examining the user block.

FILES

/dev	
/dev/kmem	kernel virtual memory
/dev/mem	memory
/dev/swap	default swap device
/dev/sxt/∗	
/dev/tty∗	
/dev/xt/∗	terminal (**tty**) names searcher files
/etc/passwd	UID information supplier
/etc/ps_data	internal data structure

SEE ALSO

kill(1), **whodo**(1M), **lseek**(2), **getpriority**(3C), **nice**(3B)

NOTES

Things can change while **ps** is running; the picture it gives is only a close approximation to the current state. Some data printed for defunct processes is irrelevant.

If no *term* or *num* is specified, **ps** checks the standard input, the standard output, and the standard error in that order, looking for the controlling terminal and will attempt to report on processes associated with the controlling terminal. In this situation, if the standard input, the standard output, and the standard error are all redirected, **ps** will not find a controlling terminal, so there will be no report.

On a heavily loaded system, **ps** may report an **lseek**(2) error and exit. **ps** may seek to an invalid user area address, having obtained the address of process' user area, **ps** may not be able to seek to that address before the process exits and the address becomes invalid.