

NAME

ps – report process status

SYNOPSIS

ps [*options*]

DESCRIPTION

ps gives a snapshot of the current processes. If you want a repetitive update of this status, use **top**. This man page documents the /proc-based version of **ps**, or tries to.

OPTIONS

This version of **ps** accepts several kinds of options.

Unix options may be grouped and must be preceded by a dash. BSD options may be grouped and must not be used with a dash. GNU long options are preceded by two dashes.

Options of different types may be freely mixed.

Set the **I_WANT_A_BROKEN_PS** environment variable to force BSD syntax even when options are preceded by a dash. The **PS_PERSONALITY** environment variable (described below) provides more detailed control of **ps** behavior.

SIMPLE PROCESS SELECTION

-A	select all processes
-N	negate selection
-a	select all with a tty except session leaders
-d	select all, but omit session leaders
-e	select all processes
T	select all processes on this terminal
a	select all processes on a terminal, including those of other users
g	really all, even group leaders (does nothing w/o SunOS settings)
r	restrict output to running processes
x	select processes without controlling ttys
--deselect	negate selection

PROCESS SELECTION BY LIST

-C	select by command name
-G	select by RGID (supports names)
-U	select by RUID (supports names)
-g	select by session leader OR by group name
-p	select by PID
-s	select processes belonging to the sessions given
-t	select by tty
-u	select by effective user ID (supports names)
U	select processes for specified users
p	select by process ID
t	select by tty
--Group	select by real group name or ID
--User	select by real user name or ID
--group	select by effective group name or ID

--pid select by process ID
--sid select by session ID
--tty select by terminal
--user select by effective user name or ID
-123 implied --sid
123 implied --pid

OUTPUT FORMAT CONTROL

-O is preloaded "-o"
-c different scheduler info for -l option
-f does full listing
-j jobs format
-l long format
-o user-defined format
-y do not show flags; show rss in place of addr
O is preloaded "o" (overloaded)
X old Linux i386 register format
j job control format
l Display long format
o specify user-defined format
s display signal format
u display user-oriented format
v display virtual memory format
--format user-defined format

OUTPUT MODIFIERS

-H show process hierarchy (forest)
-m shows threads
-n sets namelist file
-w wide output
C use raw CPU time for %CPU instead of decaying average
N specify namelist file
O sorting order (overloaded)
S,
--cumulative include some dead child process data (as a sum with the parent)
c true command name
e show environment after the command
f, --forest ASCII-art process hierarchy (forest)
h no header (or, one header per screen in the BSD personality)
m all threads
n numeric output for WCHAN and USER
--cols,
--columns,
--width set screen width
--headers repeat header lines, one per page of output
--no-headers print no header line at all
--lines,
--rows set screen height
--sort specify sorting order

INFORMATION

-V, V,
--version print version

L list all format specifiers
--help print help message
--info print debugging info

OBSOLETE

A increases the argument space (DecUnix)
M use alternate core (try -n or N instead)
W get swap info from ... not /dev/drum (try -n or N instead)
k use /vmcore as c-dumpfile (try -n or N instead)

NOTES

The **-g** option can select by session leader OR by group name. Selection by session leader is specified by many standards, but selection by group is the logical behavior that several other operating systems use. This **ps** will select by session leader when the list is completely numeric (as sessions are). Group ID numbers will work only when some group names are also specified.

The **m** option should not be used. Use **-m** or **-o** with a list. (**m** displays memory info, shows threads, or sorts by memory use)

The **h** option is problematic. Standard BSD **ps** uses the option to print a header on each page of output, but older Linux **ps** uses the option to totally disable the header. This version of **ps** follows the Linux usage of not printing the header unless the BSD personality has been selected, in which case it prints a header on each page of output. Regardless of the current personality, you can use the long options **--headers** and **--no-headers** to enable printing headers each page and disable headers entirely, respectively.

Terminals (ttys, or screens for text output) can be specified in several forms: */dev/ttyS1*, *ttyS1*, *S1*. Obsolete **ps t** (your own terminal) and **ps t?** (processes without a terminal) syntax is supported, but modern options (**T**, **-t** with list, **x**, **t** with list) should be used instead.

The BSD **O** option can act like **-O** (user-defined output format with some common fields predefined) or can be used to specify sort order. Heuristics are used to determine the behavior of this option. To ensure that the desired behavior is obtained, specify the other option (sorting or formatting) in some other way.

For sorting, BSD **O** option syntax is **O[+|-]k1[, [+|-]k2[,...]]** Order the process listing according to the multi-level sort specified by the sequence of short keys from SORT KEYS, *k1*, *k2*, ... The '+' is quite optional, merely re-iterating the default direction on a key. '-' reverses direction only on the key it precedes. The **O** option must be the last option in a single command argument, but specifications in successive arguments are catenated.

GNU sorting syntax is **--sortX[+|-]key[, [+|-]key[,...]]** Choose a multi-letter key from the SORT KEYS section. X may be any convenient separator character. To be GNU-ish use '='. The '+' is really optional since default direction is increasing numerical or lexicographic order. For example, **ps jax --sort=uid,-ppid,+pid**

This **ps** works by reading the virtual files in /proc. This **ps** does not need to be suid kmem or have any privileges to run. Do not give this **ps** any special permissions.

This **ps** needs access to a namelist file for proper WCHAN display. The namelist file must match the current Linux kernel exactly for correct output.

To produce the WCHAN field, **ps** needs to read the System.map file created when the kernel is compiled. The search path is:

\$PS_SYSTEM_MAP

```

/boot/System.map- 'uname -r'
/boot/System.map
/lib/modules/'uname -r'/System.map
/usr/src/linux/System.map

```

The member `used_math` of `task_struct` is not shown, since `cr0.s` checks to see if `math` is present. This causes the `math` flag to be set for all processes, and so it is worthless. (Somebody fix `libc` or the kernel please)

Programs swapped out to disk will be shown without command line arguments, and unless the `c` option is given, in brackets.

%CPU shows the `cputime/realtime` percentage. It will not add up to 100% unless you are lucky. It is time used divided by the time the process has been running.

The `SIZE` and `RSS` fields don't count the page tables and the `task_struct` of a `proc`; this is at least 12k of memory that is always resident. `SIZE` is the virtual size of the `proc` (`code+data+stack`).

Processes marked `<defunct>` are dead processes (so-called "zombies") that remain because their parent has not destroyed them properly. These processes will be destroyed by `init(8)` if the parent process exits.

PROCESS FLAGS

ALIGNWARN	001	print alignment warning msgs
STARTING	002	being created
EXITING	004	getting shut down
PTRACED	010	set if <code>ptrace(0)</code> has been called
TRACESYS	020	tracing system calls
FORKNOEXEC	040	forked but didn't exec
SUPERPRIV	100	used super-user privileges
DUMPCORE	200	dumped core
SIGNALLED	400	killed by a signal

PROCESS STATE CODES

D	uninterruptible sleep (usually IO)
R	runnable (on run queue)
S	sleeping
T	traced or stopped
Z	a defunct ("zombie") process

For BSD formats and when the "stat" keyword is used, additional letters may be displayed:

W	has no resident pages
<	high-priority process
N	low-priority task
L	has pages locked into memory (for real-time and custom IO)

SORT KEYS

Note that the values used in sorting are the internal values `ps` uses and not the 'cooked' values used in some of the output format fields. Pipe `ps` output into the `sort(1)` command if you want to sort the cooked values.

KEY	LONG	DESCRIPTION
c	cmd	simple name of executable

C	cmdline	full command line
f	flags	flags as in long format F field
g	pgrp	process group ID
G	tpgid	controlling tty process group ID
j	cutime	cumulative user time
J	cstime	cumulative system time
k	utime	user time
K	stime	system time
m	min_ft	number of minor page faults
M	maj_ft	number of major page faults
n	cmin_ft	cumulative minor page faults
N	cmaj_ft	cumulative major page faults
o	session	session ID
p	pid	process ID
P	ppid	parent process ID
r	rss	resident set size
R	resident	resident pages
s	size	memory size in kilobytes
S	share	amount of shared pages
t	tty	the minor device number of tty
T	start_time	time process was started
U	uid	user ID number
u	user	user name
v	vsz	total VM size in kB
y	priority	kernel scheduling priority

AIX FORMAT DESCRIPTORS

This ps supports AIX format descriptors, which work somewhat like the formatting codes of **printf(1)** and **printf(3)**. For example, the normal default output can be produced with this: **ps -eo "%p %y %x %c"**

CODE	NORMAL	HEADER
%C	pcpu	%CPU
%G	group	GROUP
%P	ppid	PPID
%U	user	USER
%a	args	COMMAND
%c	comm	COMMAND
%g	rgroup	RGROUP
%n	nice	NI
%p	pid	PID
%r	pgid	PGID
%t	etime	ELAPSED
%u	ruser	RUSER
%x	time	TIME
%y	tty	TTY
%z	vsz	VSZ

STANDARD FORMAT SPECIFIERS

These may be used to control both output format and sorting. For example: **ps -eo pid,user,args --sort user**

CODE	HEADER
%cpu	%CPU
%mem	%MEM

alarm	ALARM
args	COMMAND
blocked	BLOCKED
bsdstart	START
bsdtime	TIME
c	C
caught	CAUGHT
cmd	CMD
comm	COMMAND
command	COMMAND
cputime	TIME
drs	DRS
dsiz	DSIZ
egid	EGID
egroup	EGROUP
eip	EIP
esp	ESP
etime	ELAPSED
euid	EUID
euser	EUSER
f	F
fgid	FGID
fgroup	FGROUP
flag	F
flags	F
fname	COMMAND
fsgid	FSGID
fsgroup	FSGROUP
fsuid	FSUID
fsuser	FSUSER
fuid	FUID
fuser	FUSER
gid	GID
group	GROUP
ignored	IGNORED
intpri	PRI
lim	LIM
longname	TTY
lstart	STARTED
m_drs	DRS
m_trs	TRS
maj_ft	MAJFL
majflt	MAJFLT
min_ft	MINFL
minflt	MINFLT
ni	NI
nice	NI
nwchan	WCHAN
opri	PRI
pagein	PAGEIN
pcpu	%CPU
pending	PENDING
pgid	PGID

pgrp	PGRP
pid	PID
pmem	%MEM
ppid	PPID
pri	PRI
priority	PRI
rgid	RGID
rgroup	RGROUP
rss	RSS
rssize	RSS
rsz	RSZ
ruid	RUID
ruser	RUSER
s	S
sess	SESS
session	SESS
sgi_p	P
sgi_rss	RSS
sgid	SGID
sgroup	SGROUP
sid	SID
sig	PENDING
sig_block	BLOCKED
sig_catch	CATCHED
sig_ignore	IGNORED
sig_pend	SIGNAL
sigcatch	CAUGHT
sigignore	IGNORED
sigmask	BLOCKED
stackp	STACKP
start	STARTED
start_stack	STACKP
start_time	START
stat	STAT
state	S
stime	STIME
suid	SUID
suser	SUSER
svgid	SVGID
svgroup	SVGROUP
svuid	SVUID
svuser	SVUSER
sz	SZ
time	TIME
timeout	TMOUT
tmout	TMOUT
tname	TTY
tpgid	TPGID
trs	TRS
trss	TRSS
tsiz	TSIZ
tt	TT
tty	TT

tty4	TTY
tty8	TTY
ucmd	CMD
ucomm	COMMAND
uid	UID
uid_hack	UID
uname	USER
user	USER
vsize	VSZ
vsz	VSZ
wchan	WCHAN

ENVIRONMENT VARIABLES

The following environment variables could affect ps:

COLUMNS	Override default display width.
LINES	Override default display height.
PS_PERSONALITY	Set to one of posix,old,linux,bsd,sun,digital...
CMD_ENV	Set to one of posix,old,linux,bsd,sun,digital...
I_WANT_A_BROKEN_PS	Force obsolete command line interpretation.
LC_TIME	Date format.
PS_COLORS	Not currently supported.
PS_FORMAT	Default output format override.
PS_SYSMAP	Default namelist (System.map) location.
PS_SYSTEM_MAP	Default namelist (System.map) location.
POSIXLY_CORRECT	Don't find excuses to ignore bad "features".
UNIX95	Don't find excuses to ignore bad "features".
_XPG	Cancel CMD_ENV=irix non-standard behavior.

In general, it is a bad idea to set these variables. The one exception is **CMD_ENV** or **PS_PERSONALITY**, which could be set to Linux for normal systems. Without that setting, **ps** follows the useless and bad parts of the Unix98 standard.

PERSONALITY

390	like the S/390 OpenEdition ps
aix	like AIX ps
bsd	like FreeBSD ps (totally non-standard)
compaq	like Digital Unix ps
debian	like the old Debian ps
digital	like Digital Unix ps
gnu	like the old Debian ps
hp	like HP-UX ps
hpux	like HP-UX ps
irix	like Irix ps
linux	***** RECOMMENDED *****
old	like the original Linux ps (totally non-standard)
posix	standard
sco	like SCO ps
sgi	like Irix ps
sun	like SunOS 4 ps (totally non-standard)
sunos	like SunOS 4 ps (totally non-standard)

sysv	standard
unix	standard
unix95	standard
unix98	standard

EXAMPLES

To see every process on the system using standard syntax:

```
ps -e
```

To see every process on the system using BSD syntax:

```
ps ax
```

To see every process except those running as root (real & effective ID)

```
ps -U root -u root -N
```

To see every process with a user-defined format:

```
ps -eo pid,tt,user,fname,tmout,f,wchan
```

Odd display with AIX field descriptors:

```
ps -o "%u : %U : %p : %a"
```

Print only the process IDs of syslogd:

```
ps -C syslogd -o pid=
```

CONFORMING TO

This **ps** conforms to version 2 of the Single Unix Specification.

AUTHOR

ps was originally written by Branko Lankester <lankeste@fwi.uva.nl>. Michael K. Johnson <johnsonm@redhat.com> re-wrote it significantly to use the `proc` filesystem, changing a few things in the process. Michael Shields <mjshield@nyx.cs.du.edu> added the `pid-list` feature. Charles Blake <cblake@bbn.com> added multi-level sorting, the `dirent`-style library, the device name-to-number `mmaped` database, the approximate binary search directly on `System.map`, and many code and documentation cleanups. David Mossberger-Tang wrote the generic BFD support for `psupdate`. Albert Cahalan <acahalan@cs.uml.edu> rewrote **ps** for full Unix98 and BSD support, along with some ugly hacks for obsolete and foreign syntax.

Please send bug reports to <acahalan@cs.uml.edu> or use the Debian Bug Tracking System.

SEE ALSO

top(1), **pstree(1)**, **proc(5)**