

PS

NOMBRE

ps - informa del estado de los procesos

SINOPSIS

ps [-] [**l****u****j****s****v****m****a****x****S****c****e****w****h****r****n****u**] [**t****x****x**] [**0**[**+****-**]**k****1**[**+****-**]**k****2**...]] [**p****i****d****s**]

también existen tres opciones largas:

--sort**X**[**+****-**]**key**[,**+****-**]**key**[,...]]

--help

--version

Se están desarrollando más opciones largas...

DESCRIPCIÓN

ps muestra una instantánea de los procesos actuales. Si quiere una actualización continua, use **top**. Esta página de manual documenta (o por lo menos intenta documentar) la versión de **ps** basada en [/proc](#).

OPCIONES DE LA LÍNEA DE COMANDOS

Las opciones de la línea de comandos para esta versión de **ps** proceden de la versión BSD de **ps**, no de la versión System V.

Es recomendable que los argumentos de la línea de comandos **no** estén precedidos por un carácter **`-'**, porque en un futuro, el **`-'** se utilizará para indicar argumentos del estándar Unix98, mientras que sin **`-'** indicará el modo actual **``BSD extendido''**.

Por ahora, **ps** mostrará una advertencia (warning) si usa un **`-'** para una opción corta, pero aún así, todavía funcionará. Si tiene scripts shell que utilizan los argumentos de **ps** tipo BSD, preste atención a las advertencias y solucíonelas, o sus scripts dejarán de funcionar de forma correcta en un futuro. Si quiere deshabilitar estas advertencias, establezca la variable de entorno **I_WANT_A_BROKEN_PS**.

Existen además algunas **``opciones largas''** estilo GNU; véase más abajo.

l	formato largo
u	formato usuario: muestra el usuario y la hora de inicio
j	formato trabajo (jobs): pgid sid
s	formato señal (signal)
v	formato vm
m	muestra información de memoria (combínese con p para obtener el número de páginas).
f	formato "forest" ("bosque") de familias en forma de árbol
a	muestra también los procesos de otros usuarios
x	muestra procesos que no están controlados por ninguna terminal
S	añade tiempo de CPU y fallos de página de los hijos
c	nombre del comando obtenido de task_struct
e	muestra ambiente (environment) después del nombre del comando y `

+ '

w Salida ancha (wide): no se truncan las líneas de comando para que quepan en una línea. Para ser exactos, cada **w** que se especifica añadirá una posible línea a la salida. Si el espacio no se necesita, no se usa. Puede usar hasta 100 **w**'s.

h sin cabecera (header)

r sólo procesos que se están ejecutando

n salida numérica para **USER** y **WCHAN**.

t~~xx~~ sólo procesos controlados por la tty **~~xx~~**; para **~~xx~~** debe usar bien el nombre de un dispositivo bajo **"/dev"** o bien ese nombre sin las letras **tty** o **cu** que lo preceden. Esta es la operación inversa que **ps** utiliza para imprimir el nombre abreviado de tty en el campo **TT**, por ejemplo, **ps -t1**.

O[+|-]k1[, [+|-]k2[, ...]]
 Ordena la lista de procesos de acuerdo con el ordenamiento multi-nivel especificado por la secuencia de claves **ordenación** de **CLAVES DE ORDENACIÓN**, **k1**, **k2**, Existen especificaciones de ordenación por defecto para cada uno de los formatos de **ps**. Éstas pueden ser anuladas por una ordenación especificada por el usuario. El **`+'** es opcional, meramente para reiterar la dirección por defecto de una clave. **`-'** invierte la dirección sólo de la clave que precede. Como con **t** y **pids**, la opción **O** debe ser la última opción en un argumento, aunque las especificaciones en argumentos posteriores son concatenadas.

pids Lista sólo los procesos especificados; están delimitados por comas. La lista se debe dar inmediatamente después de la última opción en un argumento simple, sin intervención de espacios, por ejemplo **ps -j1,4,5**. Las listas especificadas en los argumentos siguientes son concatenadas, por ejemplo **ps -l 1,2 3,4 5 6** listará todos los procesos del 1 al 6 en formato largo. Los pids se listan incluso si contradicen a las opciones **'a'** y **'x'**

OPCIONES DE LA LÍNEA DE COMANDOS LARGAS

Estas opciones están precedidas por un doble guión.

--sort~~X~~[+|-]key[, [+|-]key[, ...]]
 Selecciona una clave de varias letras de la sección **CLAVES DE ORDENACIÓN**. **X** puede ser cualquier carácter de separación. GNU prefiere **`=**'. El **`+'** es realmente opcional, ya que la dirección por defecto es creciente en orden numérico o lexicográfico. Por ejemplo: **ps -jax --sort=uid, -ppid, +pid**

--help
 Muestra un mensaje de ayuda que resume el uso y da una lista de claves de ordenación soportadas. Esa lista puede estar más actualizada que la de esta página del manual.

--version
 Muestra la versión y la procedencia de este programa.

CLAVES DE ORDENACIÓN

Nótese que los valores utilizados en la ordenación son valores internos que **ps** utiliza y no los valores **`transformados'** (**`cooked'**) que aparecen en alguno de los campos de la salida. Si alguien, voluntariamente, quiere escribir alguna función especial de comparación para los valores transformados, ... ;-)

CORTA	LARGA	DESCRIPCIÓN
c	cmd	nombre simple del ejecutable
C	cmdline	línea de comandos completa
f	flags	flags como en el campo F del formato largo
g	pgrp	ID del grupo del proceso
G	tpgid	ID del grupo del proceso que controla la tty
j	cutime	tiempo de usuario acumulado
J	cstime	tiempo de sistema acumulado
k	utime	tiempo de usuario
K	stime	tiempo de sistema
m	min_flt	número de fallos de página menores
M	maj_flt	número de fallos de página mayores
n	cmin_flt	fallos menores de página acumulados
N	cmaj_flt	fallos mayores de página acumulados
o	session	ID de la sesión
p	pid	ID del proceso
P	ppid	ID del proceso padre
r	rss	tamaño de la parte residente
R	resident	páginas residentes
s	size	memoria usada en kilobytes
S	share	número de páginas compartidas
t	tty	el tty que usa el proceso
T	start_time	hora en la que el proceso se inició
U	uid	ID del usuario
u	user	nombre del usuario
v	vsize	tamaño total VM en bytes
y	priority	prioridad en el planificador del kernel

DESCRIPCIÓN DE LOS CAMPOS

PRI Este es el campo contador de la estructura de la tarea. Es el tiempo en **HZ** de la posible rodaja de tiempo del proceso.

NI Valor nice estándar de Unix; un número positivo significa menos tiempo de cpu.

SIZE Tamaño virtual de la imagen; Tamaño de text+data+stack.

RSS Tamaño de la parte residente; kilobytes del programa en memoria.

WCHAN

Nombre de la función del kernel que el proceso estaba ejecutando cuando pasó a estar durmiendo, con el ``sys_`` quitado del nombre de la función. Si no existe el fichero /etc/psdatabase, se muestra sólo un número hexadecimal.

STAT Información acerca del estado del proceso. El primer campo es **R** para preparado para ejecución (runnable), **S** para durmiendo (sleeping), **D** para indicar letargo ininterrumpible (uninterruptible sleep), **T** para parado o trazado (traced), o **Z** para un proceso zombie. El segundo campo contiene **W** si el proceso no tiene páginas residentes. El tercer campo es **N** si el proceso tiene un número nice positivo (campo NI).

TT Tty asociada.

PAGEIN

Número de fallos de página mayores (fallos de página que causan lectura de páginas desde el disco, incluyendo las páginas leídas desde el buffer de caché).

TRS Tamaño del código residente.

SWAP Kilobytes (o páginas si usamos **-p**) en el dispositivo de intercambio (swap).

SHARE
Memoria compartida.

ACTUALIZANDO

Este **ps** (basado en **proc**) lee los ficheros en el sistema de ficheros **proc** , montado en [/proc](#). No necesita un **kmem** con **suid** ni privilegios para funcionar. No ~~da~~ a este ~~ps~~ ningún ~~permiso~~ especial.

Necesitará actualizar el fichero [/etc/psdatabase](#) ejecutando [/usr/sbin/psupdate](#) para obtener una información comprensible del campo **WCHAN**. Debería hacer esto cada vez que compile un nuevo kernel. Debería también ejecutar 'ps' como root una vez y cada vez que los dispositivos tty en el directorio "[/dev](#)" cambien.

Como **procps-1.00**, **ps/top** leen **System.map** directamente si existe. El camino de búsqueda para la resolución de dirección a símbolo del kernel (address-to-symbol) es:

```
$PS_SYSTEM_MAP
/boot/System.map-`uname -r`
/boot/System.map
/lib/modules/`uname -r`/System.map
/etc/psdatabase
/boot/psdatabase-`uname -r`
/boot/psdatabase,
/lib/modules/`uname -r`/psdatabase
```

NOTAS

El miembro **used_math** de **task_struct** no se muestra, ya que **crt0.s** comprueba si existe emulación de coprocesador matemático. Esto hace que el flag **math** esté activado para todos los procesos, por lo que es inútil.

Los programas intercambiados a disco se mostrarán sin los argumentos de la línea de comando, y a no ser que se de la opción **c** , entre paréntesis.

%CPU muestra el porcentaje tiempo de cpu/tiempo real. No llegará al 100% a no ser que tengas suerte. Es el tiempo utilizado dividido por el tiempo que el proceso se ha estado ejecutando.

Los campos **SIZE** y **RSS** no cuentan las tablas de páginas y la **task_struct** del proceso; esto supone por lo menos 12k de memoria que siempre está residente. **SIZE** es el tamaño virtual del proceso (código+datos+pila).

Para conseguir la correspondencia entre número y nombre de dispositivo, **ps** mantiene un fichero llamado "[/etc/psdevtab](#)" (actualizado cada vez que "[/dev](#)" se actualiza y los permisos permitan la actualización). Si los permisos no lo permiten, cada invocación de **ps** requiere que se ejecute un [stat\(2\)](#) a cada fichero en el directorio "[/dev](#)". Si las entradas en "[/dev](#)" cambian a menudo en su sistema, debería ejecutar **ps** como root a menudo. Puede que añada un fichero accesorio bajo **\$HOME** si se demanda por los usuarios.

AUTOR

ps fue escrito por primera vez por Branko Lankester <lankeste@fwi.uva.nl>. Michael K. Johnson <johnsonm@redhat.com> lo modificó de forma significativa para que utilizara el sistema de ficheros proc, cambiando algunas cosas en el proceso. Michael Shields <mjshield@nyx.cs.du.edu> añadió la posibilidad de dar listas de pids. Charles Blake <cblake@bbn.com> añadió ordenación multi-nivel, la librería estilo dirent, la base de datos de correspondencia de nombre de dispositivo a número, la búsqueda binaria aproximada en System.map, y mejoró el código y la documentación. David Mossberger-Tang escribió el soporte genérico BFD para psupdate. Michael K. Johnson <johnsonm@redhat.com> es el encargado del mantenimiento actualmente.

Por favor, envíe informes de los fallos a <procps-bugs@redhat.com>