**Rocío García Núñez y Rubén Izquierdo Belinchón**

**-Comandos procfs**

* cpuinfo: Este es una colección de elementos dependientes de la CPU y de  
   la arquitectura del sistema. Para cada arquitectura soportada,  
   una lista diferente. Dos entradas comunes son processor que da  
   el número de CPU y bogomips, una constante del sistema que se  
   calcula durante el arranque del núcleo. Las máquinas SMP tienen  
   información para cada CPU.
* meminfo

Éste es usado por free(1) para informar de la cantidad de

memoria libre y usada en el sistema (tanto física como de

intercambio) así como de la memoria compartida y los buffers

usados por el núcleo.

El formato es el mismo que el de free(1), salvo que los datos se

dan en bytes y no en KB.

* swaps Áreas de intercambio en uso. Véase también swapon(8).
* loadavg

Los números de carga media dan el número promedio de trabajos en

la cola de ejecución (estado R) o en espera de E/S de disco

(estado D) en los últimos 1, 5 y 15 minutos. Estos números son

idénticos a los números de carga media dados por uptime(1) y

otros programas.

* diskstats :Este archivo contiene las estadísticas I/O para cada disco
* vmstat: Este archivo muestra varias estadísticas en memoria virtual.Cada línea del archivo contiene un par nombre valor delimitado por espacios algunas líneas solo aparecen si el kernel ha sido configurado con las opciones adecuadas. (En algunos casos , las opciones requeridas para archivos particulares han cambiado con las versiones del kernel , así que no están listados aquí.)
* interrupts: Guarda el nº de interrupciones de CPU por dispositivos de I/O
* $$/status: información sobre el estado del proceso
* $$/maps : UN archivo que contiene las actuales regiones de memoria mapeadas y sus permisos de acceso
* $$/limits Este archivo muestra las restricciones del proceso
* $$/sched Información sobre la planificación del proceso actual
* $$/io Contiene las estadísticas de I/O del proceso
* $$/net/dev :El pseudo-archivo contiene la información del estado de red del proceso
* $$/net/netstat: Archivo que contiene estadísticas del protocolo TCP/IP del proceso

* **Time**

time -p

usuario@debian:~$ time -p ls

Descargas Documentos Escritorio Imágenes Música Plantillas Público Vídeos

real 0.13

user 0.00

sys 0.00

usuario@debian:~$ time -p ls -l

total 32

drwxr-xr-x 2 usuario usuario 4096 oct 18 2014 Descargas

drwxr-xr-x 2 usuario usuario 4096 oct 18 2014 Documentos

drwxr-xr-x 2 usuario usuario 4096 oct 18 2014 Escritorio

drwxr-xr-x 2 usuario usuario 4096 oct 18 2014 Imágenes

drwxr-xr-x 2 usuario usuario 4096 oct 18 2014 Música

drwxr-xr-x 2 usuario usuario 4096 oct 18 2014 Plantillas

drwxr-xr-x 2 usuario usuario 4096 oct 18 2014 Público

drwxr-xr-x 2 usuario usuario 4096 oct 18 2014 Vídeos

real 0.18

user 0.00

sys 0.00

Gracias a time y a la opción -p, que nos hace más legible el tiempo, se aprecia que tarda más al añadir la opción -l ya que se listan los permisos y demás información.

User time (seconds) → struct timeval ru\_utime

System time (seconds) → struct timeval ru\_stime

Percent of CPU this job got →

Elapsed (wall clock) time (h:mm:ss or m:ss):

Average shared text size (kbytes):

Average unshared data size (kbytes) → ru\_idrss;

Average stack size (kbytes) → ru\_isrss;

Average total size (kbytes):

Maximum resident set size (kbytes) → long ru\_maxrss;

Average resident set size (kbytes):

Major (requiring I/O) page faults → ru\_majflt;

Minor (reclaiming a frame) page faults → ru\_minflt;

Voluntary context switches → ru\_nvcsw

Involuntary context switches → ru\_nivcsw

Swaps → ru\_nswap

File system inputs:

File system outputs:

Socket messages sent:

Socket messages received:

Signals delivered:

Page size (bytes): PAGE\_SIZE

Exit status:

time - p find /usr > /dev/null

Observamos que la primera vez que accedemos tarda más que las siguientes veces que tarda menos en acceder

usuario@debian:~$ time -p dd if=/dev/zero of=/var/tmp/prueba count=100K bs=1K

102400+0 registros leídos

102400+0 registros escritos

104857600 bytes (105 MB) copiados, 0,129242 s, 811 MB/s

real 0.18

user 0.00

sys 0.10

usuario@debian:~$ time -p dd if=/dev/zero of=/var/tmp/prueba count=100K bs=1K oflag=direct

102400+0 registros leídos

102400+0 registros escritos

104857600 bytes (105 MB) copiados, 4,80166 s, 21,8 MB/s

real 4.81

user 0.00

sys 1.19

* **PS**

ps -U root -u root -o uname,pri,pcpu,vsz,rss k -rss

* **TOP**

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

2454 usuario 20 0 181m 10m 6940 S 0,3 1,0 0:07.86 xfce4-terminal

3415 root 20 0 0 0 0 R 0,3 0,0 0:00.09 kworker/0:3

1 root 20 0 10652 192 172 S 0,0 0,0 0:00.39 init

2 root 20 0 0 0 0 S 0,0 0,0 0:00.00 kthreadd

3 root 20 0 0 0 0 S 0,0 0,0 0:00.17 ksoftirqd/0

6 root rt 0 0 0 0 S 0,0 0,0 0:00.00 migration/0

7 root rt 0 0 0 0 S 0,0 0,0 0:00.06 watchdog/0

8 root 0 -20 0 0 0 S 0,0 0,0 0:00.00 cpuset

9 root 0 -20 0 0 0 S 0,0 0,0 0:00.00 khelper

10 root 20 0 0 0 0 S 0,0 0,0 0:00.00 kdevtmpfs

11 root 0 -20 0 0 0 S 0,0 0,0 0:00.00 netns

12 root 20 0 0 0 0 S 0,0 0,0 0:00.01 sync\_supers

13 root 20 0 0 0 0 S 0,0 0,0 0:00.00 bdi-default

14 root 0 -20 0 0 0 S 0,0 0,0 0:00.00 kintegrityd

15 root 0 -20 0 0 0 S 0,0 0,0 0:00.00 kblockd

17 root 20 0 0 0 0 S 0,0 0,0 0:00.00 khungtaskd

18 root 20 0 0 0 0 S 0,0 0,0 0:00.77 kswapd0

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

3488 usuario 20 0 1206m 410m 428 R 63,8 40,9 0:01.92 progra

Al principio kswapd0 tiene un consumo de CPU 0 , tras ejecutar el programa “cpu\_mem” va aumentando el uso de CPU y de memoria

Cuando el porcentaje del uso de la memoria es elevado entra el proceso kswapd0

Cuando el porcentaje de memoria es cercano al 100% finaliza el proceso

