LinuxTotal.com.mx

Información y servicios en Linux y Open Source

MÚLTIPLES FORMAS DE VER INFORMACIÓN DEL SISTEMA

autor: sergio.gonzalez.duran@gmail.com

Sistemas basados en GNU/Linux (Al igual que sus parientes basados en Unix como BSD o los de Macinstosh) conservan la tradición de tener multitud de comandos que permiten conocer el estado del sistema. Es decir, cada uno da pequeñas piezas de información sobre multitud de partes diferentes de lo que esta sucediendo en tu sistema Linux. Algunos de estos comandos pueden ser ejecutados por cualquier usuario y otros varios solo por root. En esta ocasión te presento, sin ningún orden en específico, una recopilación de los más útiles y usados de estos comandos.

uname

Imprime información del sistema

(Procesador instalado en el equipo)
```
#> uname -p
Intel(R) Core(TM) Duo CPU T2450  @ 2.00GHz
```

(versión del kernel)
```
#> uname -r
2.6.22.9-laptop-1mdv
```

(o toda la información de uname a través de la opción *-a*)
```
#> uname -a
Linux segolap 2.6.22.9-laptop-1mdv #1 SMP Thu Sep 27 04:17:10 CEST 2007 i686 Intel(R) Core(TM) Duo CPU T2450  @ 2.00GHz GNU/Linux
```

Este último, muestra en orden, el tipo de kernel, el nombre del equipo, versión de kernel, fecha y hora, arquitectura del CPU (i686), tipo de procesador y tipo de sistema operativo (GNU/Linux).

fdisk

Permite manipular/crear particiones en Linux, pero tiene una interesante opción de consulta, *-l*:

```
#> fdisk -l

Disk /dev/sda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0002ecbc

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          63      506016   83  Linux
/dev/sda2              64       10261    81915435   83  Linux
/dev/sda3           10262       18929    69625710   83  Linux
/dev/sda4           18930       19457     4241160    5  Extended
/dev/sda5           18930       19457     4241128+  82  Linux swap / Solaris

Disk /dev/sdb: 40.0 GB, 40007761920 bytes
255 heads, 63 sectors/track, 4864 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000063b0

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        4863    39062016    c  W95 FAT32 (LBA)
```

Podemos observar en este listado varios aspectos muy útiles, primero que tenemos dos dispositivos conectados al sistema, /dev/sda y /dev/sdb, los dos son discos duros, el primero es el propio del equipo y tiene varias particiones, incluso determinamos cual es la partición de arranque que es /dev/sda1. El segundo dispositivo contiene una partición Windows como podemos ver en la columna 'System' del último renglón 'W95 FAT32', que indiscutiblemente es de Windows, no es una memoria flash por el tamaño (Disk /dev/sdb: 40.0 GB) mostrado. Asi que se trata de un disco duro externo.

free

¿Sientes tu sistema demasiado lento?, comienza checando con **`free`** que despliega como se encuentra de saturada la memoria física RAM y la de la partición SWAP.

(la opción *-m* muestra el listado en megas)
```
#> free -m
             total       used       free     shared    buffers     cached
Mem:          2018        989       1028          0         39        450
-/+ buffers/cache:        500       1517
Swap:         4141          0       4141
```

La línea 'Mem:' es la memoría fisica RAM, que en este ejemplo tiene 2 GB de los cuáles se están usando 989 megas, bastante razonable todavía, la línea 'Swap:' muestra la partición de swap (lo que en Windows se le conoce como archivo de intercambio), que generalmente se establece al doble de la RAM y que idealmente no debe estar usada, como el ejemplo lo muestra. Cuando tu línea Swap muestra demasiado uso y casi nada libre, tienes serios problemas de rendimiento, considera entonces en incrementar tu RAM. Prueba con **`free -mt`** para ver una línea más al final con la suma de las dos Mem + Swap.

mount

Comando que se utiliza para montar dispositivos, algo complejo y con múltiples opciones. Pero para este tutorial, basta con que lo invoques sin

opción alguna ni argumentos, para que nos revele la información de que tienes montado y en que lugar esta montado.

```
#> mount
/dev/sda1 on /boot type ext3 (rw,noatime)
/dev/sda2 on / type ext3 (rw,noatime)
/dev/sda3 on /home type ext3 (rw,noatime)
none on /proc type proc (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/dev/sdb1 on /media/hd type vfat (rw,nosuid,nodev,sync,users,umask=0022,iocharset=utf8)
```

Un pequeño análisis me permite determinar que en el equipo hay tres particiones sobre el mismo disco duro (dispositivo /dev/sda), que son /boot (sda1), / (sda2), y /home (sda3), todas son del tipo 'ext3' el filesystem por defecto de Linux. Hay dos sistemas virtuales montados en /proc y otro dispositivo (/dev/sdb1) accesible a través del directorio /media/hd y que es del tipo DOS FAT. Como podrás observar, esta información se complementa a la arrojada por **`fdisk -l`**.

lsmod

Muestra el status de los módulos del kernel actualmente cargados en el sistema.

```
[root@segolap ~]# lsmod
Module                  Size  Used by
fat                    45852  1 vfat
i915                   22688  3
drm                    72628  4 i915
vmnet                  34564  16
parport_pc             32004  0
parport                31592  1 parport_pc
blkcipher               5860  1 ecb
snd_seq_midi_event      6912  1 snd_seq_oss
snd_seq                46800  5 snd_seq_dummy,snd_seq_oss,snd_seq_midi_event
snd_seq_device          7276  3 snd_seq_dummy,snd_seq_oss,snd_seq
ieee80211              31752  1 ipw3945
ieee80211_crypt         5248  2 ieee80211_crypt_wep,ieee80211
mmc_core               23108  2 mmc_block,sdhci
agpgart                27656  3 drm,intel_agp
snd_pcm                69636  3 snd_pcm_oss,snd_hda_intel
libata                108688  2 ata_piix,ahci
scsi_mod              124972  6 usb_storage,sr_mod,sg,scsi_wait_scan,sd_mod,libata
... listado no completo
```

El listado se autoexplica, el módulo, su tamaño y quien lo usa. Por ejemplo, el módulo 'ieee80211' es utilizado por el driver para tarjetas inalámbricas 'ipw3945', etc. Este comando se complementa con el de inserción de módulos **`insmod`** y con el que remueve módulos **`rmmod`**.

lspci

Lista los dispositivos PCI del sistema.

```
#> lspci
00:00.0 Host bridge: Intel Corporation Mobile 945GM/PM/GMS, 943/940GML and 945GT Express Memory Controller Hub (rev 03)
00:02.1 Display controller: Intel Corporation Mobile 945GM/GMS/GME, 943/940GML Express Integrated Graphics Controller (rev 03)
00:1b.0 Audio device: Intel Corporation 82801G (ICH7 Family) High Definition Audio Controller (rev 02)
00:1c.0 PCI bridge: Intel Corporation 82801G (ICH7 Family) PCI Express Port 1 (rev 02)
00:1d.1 USB Controller: Intel Corporation 82801G (ICH7 Family) USB UHCI Controller #2 (rev 02)
00:1d.7 USB Controller: Intel Corporation 82801G (ICH7 Family) USB2 EHCI Controller (rev 02)
00:1e.0 PCI bridge: Intel Corporation 82801 Mobile PCI Bridge (rev e2)
00:1f.0 ISA bridge: Intel Corporation 82801GBM (ICH7-M) LPC Interface Bridge (rev 02)
00:1f.1 IDE interface: Intel Corporation 82801G (ICH7 Family) IDE Controller (rev 02)
00:1f.2 SATA controller: Intel Corporation 82801GBM/GHM (ICH7 Family) SATA AHCI Controller (rev 02)
00:1f.3 SMBus: Intel Corporation 82801G (ICH7 Family) SMBus Controller (rev 02)
04:00.0 Ethernet controller: Broadcom Corporation NetLink BCM5787M Gigabit Ethernet PCI Express (rev 02)
05:00.0 Network controller: Intel Corporation PRO/Wireless 3945ABG Network Connection (rev 02)
06:00.0 FLASH memory: ENE Technology Inc ENE PCI Memory Stick Card Reader Controller
06:00.1 Generic system peripheral [0805]: ENE Technology Inc ENE PCI SmartMedia / xD Card Reader Controller
06:00.3 FLASH memory: ENE Technology Inc ENE PCI Secure Digital / MMC Card Reader Controller
```

Tomemos una línea de ejemplo:

05:00.0 Network controller: Intel Corporation PRO/Wireless 3945ABG Network Connection (rev 02)

El primer campo (05:00.0) es el slot PCI donde se ubica el dispositivo bus 05 dispositivo 00 función 0, después sigue la clase de dispositivo (Network controller), el fabricante (Intel Corporation), el nombre del dispositivo (PRO/Wireless 3945ABG Network Connection) y el número de revisión del mismo (rev 02).

Información bastante útil, ya que por ejemplo, en mi caso, este dispositivo no funcionó cuando recien instalé Linux, pero con esta info del sistema comencé a determinar el tipo de drivers que necesitaba para hacerla funcionar.

Puedes obtener aun más información de cada dispositivo PCI con la opción *-v* y aun más con **`-vv`**, asi que trata con **`lspci -vv`** y observa cuanto puedes lograr saber de cada dispositivo.

lsusb

Lista los dispositivos usb del sistema.

```
#> lsusb
Bus 005 Device 004: ID 05e3:0702 Genesys Logic, Inc. USB 2.0 IDE Adapter
Bus 005 Device 003: ID 064e:a101 Suyin Corp.
Bus 005 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 002 Device 004: ID 062a:0003 Creative Labs
Bus 002 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 004 Device 001: ID 0000:0000
```

mmmmm, no muy informativo que digamos, pero solo hay que saber buscar, asi que si usamos la opción *-v*, nos devuelve más información, en mi caso, la cámara web de mi laptop no funcionaba, para buscar los drivers o configuración adecuada busqué con este comando y encontré lo siguiente:

```
#> lsusb -v
...
Bus 005 Device 003: ID 064e:a101 Suyin Corp.
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass          239 Miscellaneous Device
  bDeviceSubClass         2 Common Class
  bDeviceProtocol         1 Interface Association
  bMaxPacketSize0        64
  idVendor           0x064e Suyin Corp.
  idProduct          0xa101
  bcdDevice            1.00
  iManufacturer           2 SuYin
  iProduct                1 Acer CrystalEye webcam
  iSerial                 3 CN0314-OV03-VA-R02.00.00
...
```

El listado es bastante largo, asi que lo muestro con lo relevante solamente, en el 'Bus 005 Device 003:' se encuentra algo llamado 'Suyin Corp', y viendo más detalle con *-v* encuentro que es 'Acer CrystalEye webcam', asi que con esto se facilita la búsqueda en Internet para conseguir los drivers adecuados para linux, cosa que con paciencia eventualmente se logra. Ya que encontré que con la distro Mandriva viene soportada por defecto.

blkid

Block Id. Despliega los atributos del dispositivo de bloque.

```
#> blkid
/dev/sda1: UUID="d22801c6-85ca-11dc-849e-afde43df714c" SEC_TYPE="ext2" TYPE="ext3"
/dev/sda2: UUID="ae22f1dc-85ca-11dc-acbd-cb4aee4dedb7" SEC_TYPE="ext2" TYPE="ext3"
/dev/sda3: UUID="d3990398-85ca-11dc-aab5-4d80db2607e2" SEC_TYPE="ext2" TYPE="ext3"
/dev/sda5: TYPE="swap" UUID="f6bfa9b2-85ca-11dc-abd6-01935478454b"
/dev/sdb1: LABEL="SEGO" UUID="46CD-5C01" TYPE="vfat"
```

dmidecode y lshw

Ahora bien, que si de determinar el hardware del equipo se trata, nada como este comando. Que lo que hace es leer la información del BIOS directamente y te regresa un listado muy completo de todo el hardware encontrado en el equipo. DMI es por Desktop Management interface y lee la información del llamado SMBIOS (System Management BIOS).

dmidecode por defecto ofrece un listado bastante largo y completo, asi que si deseas uno más corto o resumido, úsalo con *-q*.

Si no tienes instalado **dmidecode** prueba con **lshw** que básicamente hace lo mismo.

df

Reporta el uso de espacio en los discos duros.

```
# df
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda2              77G   16G   58G  22% /
/dev/sda1             479M   21M  433M   5% /boot
/dev/sda3              66G   36G   30G  55% /home
/dev/sdb1              38G   24G   14G  64% /media/hd
```

Muy fácil de entender y usar, úsalo seguido, sobre todo si descargas bastante y asi podrás saber cuando se están llenando tus dispositivos de almacenamiento. En algunas versiones de **df** tendrás que usar la opción *-h* (formato humano) para que puedas ver el mismo listado mostrado en Megas o Gigas.

uptime

Muestra cuanto tiempo lleva prendido el sistema y otra información.

```
#> uptime
 19:59:45 up  2:18,  2 users,  load average: 1.14, 1.13, 1.09
```

Primero la hora actual, seguido de 'up 2:18', que significa prendido por dos horas y 18 minutos, claro este campo puede cambiar a días, etc., dos usuarios en el sistema y por último la carga promedio del CPU (load average), en el último minuto, 5 y 15 respectivamente. Mientras más bajo este número es mejor, queriendo decir que por ejemplo, se requieren 1.14 procesadores en el momento que se ejecutó 'uptime' para en ese preciso instante terminar con todos los procesos del sistema. Esto no es exactamente preciso pero te puede dar una buena idea lo cargado o desocupado que esta tu CPU. Ahora bien, ¿quienes son esos dos usuarios en el sistema?, veámoslo con el siguiente comando.

w

Muestra que usuarios están en el sistema y lo que están haciendo.

```
# w
 20:07:12 up  2:25,  2 users,  load average: 1.18, 1.12, 1.09
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1      19:09    7:34   0.16s  0.16s -bash
sergio   :0        17:43   ?xdm?   2:22m  0.06s /bin/sh /usr/bin/quanta
```

La primera línea de **w** es lo mismo que regresa **uptime**, y después nos dice quienes son los dos usuarios en el sistema, en que terminal están 'TTY', si fuera desde otro equipo mostraría la IP, la hora en que se loguearon 'LOGIN@', y la última columna muestra lo que están ejecutando en el momento en que se ejecutó **w**.

Como complemento de **uptime** y **w** puedes usar lo siguiente:

```
#> who -b
  system boot  2008-01-13 17:41
```

Indica la fecha y hora en que el sistema inició.

lsof

List open files. Muestra los archivos que un proceso ha abierto para poder ejecutarse.

```
(ejecutamos 'man lsof' en una terminal)
#> man lsof
(desde otra terminal determinamos su PID)
#> ps -ed | grep man
root      9700  6514  0 21:11 pts/1    00:00:00 man lsof
(y ejecutamos lsof con la opción -p)
#> lsof -p 9700
COMMAND  PID USER   FD   TYPE DEVICE    SIZE    NODE NAME
man     9700 root  cwd    DIR    8,2    4096 5603329 /root
man     9700 root  rtd    DIR    8,2    4096       2 /
man     9700 root  txt    REG    8,2   43416 9529630 /usr/bin/man
man     9700 root  mem    REG    8,2  254076 9520457 /usr/share/locale/UTF-8/LC_CTYPE
man     9700 root  mem    REG    8,2 1298800 7277050 /lib/i686/libc-2.6.1.so
man     9700 root  mem    REG    8,2      52 9521060 /usr/share/locale/en_US.UTF-8/LC_MESSAGES/SYS_LC_MESSAGES
man     9700 root  mem    REG    8,2   26052 9519467 /usr/lib/gconv/gconv-modules.cache
man     9700 root  mem    REG    8,2  565473 7274508 /lib/ld-2.6.1.so
man     9700 root    0u   CHR  136,1               3 /dev/pts/1
man     9700 root    1u   CHR  136,1               3 /dev/pts/1
man     9700 root    2u   CHR  136,1               3 /dev/pts/1
man     9700 root    3r   REG    8,2    4808 4523300 /etc/man.config
```

Podemos observar de las librerias, archivos de configuración (última línea), y los comandos que se invocaron para ejcutar una consulta de manual. este comando **lsof** es altamente útil cuando se trata de determinar las dependencias que un programa requiere para ejecutarse.

Si utilizas **lsof** sin argumentos te dará un larguísimo listado de todos los procesos que se estén ejecutando en ese momento.

last y lastb

last muestra un listado de los últimos usuarios logueados al sistema e información relevante, **lastb** last bad, muestra los últimos intentos de logueo al sistema que fracasaron, utilísimo para determinar posibles intentos de acceso ilegítimo al sistemo (hackeo).

```
#> last
root     tty1                         Sun Jan 13 19:59   still logged in
sergon   :0                           Sun Jan 13 17:43   still logged in
reboot   system boot  2.6.22.9-laptop- Sun Jan 13 17:41          (04:19)
root     tty1                         Sun Jan 13 00:23 - crash  (17:18)
sergon   :0                           Sat Jan 12 23:56 - 00:48  (00:52)
reboot   system boot  2.6.22.9-laptop- Sat Jan 12 23:55          (22:05)
sergon   :0                           Sat Jan 12 19:35 - down   (00:57)
reboot   system boot  2.6.22.9-laptop- Sat Jan 12 19:34          (00:59)
sergon   :0                           Sat Jan 12 17:41 - down   (01:16)
reboot   system boot  2.6.22.9-laptop- Sat Jan 12 17:40          (01:17)
sergon   :0                           Sat Jan 12 08:15 - 12:41  (04:25)
reboot   system boot  2.6.22.9-laptop- Sat Jan 12 08:15          (04:26)
sergon   :0                           Fri Jan 11 22:11 - crash  (10:03)
reboot   system boot  2.6.22.9-laptop- Fri Jan 11 21:49          (14:51)
sergon   :0                           Thu Jan 10 22:12 - 22:36  (00:23)
reboot   system boot  2.6.22.9-laptop- Thu Jan 10 22:11          (00:24)
```

Podemos ver que usuario se logueó, en que terminal, día, fecha y hora, a que hora terminó o si continua logueado (still logged in). Es posible también conocer por ejemplo en las líneas que dice 'crash' que el sistema no se apagó adecuadamente.

```
#> lastb
# lastb
pedro    192.168.0.10                 Sun Jan 13 22:04 - 22:04  (00:00)
root     tty2                         Sun Jan 13 21:20 - 21:20  (00:00)
```

Con **lastb** obtenemos los intentos de logueo que fracasaron. Por ejemplo, en un sistema real en producción donde no existiera el usuario 'pedro' resultaría obvio que alguien esta tratando de obtener acceso remoto, adivinando usuario:contraseña. Deberías preocuparte enormemente y tomar acción, si en el listado de **last** observas un logueo de root u otro usuario que tu como administrador sepas no debió entrar al sistema en esas fechas u horas, o peor aun que se trata de tu ¡¡usuario!! y no habías ingresado previamente. Con seguridad significa que ya te hackearon tu sistema o consiguieron tu contraseña.

dmesg

Parte del servidor de mensajes del sistema syslog, **dmesg** es principalmente usado para mostrar los mensajes que se mostraron en pantalla cuando se inicio (boot) el sistema. Se usa sobretodo para realizar depuraciones al sistema de como se están cargando los diversos módulos y componentes al arranque del sistema o ya en ejecución. Debido a lo extenso del sistema, es conveniente redireccionar la salida a un archivo:

```
#> dmesg > mensajes
```

Con **less** o **cat** o en tu editor favorito puedes con calma analizar el archivo.

ps

El comando por excelencia para mostrar información de procesos, en este artículo de LinuxTotal.com.mx se encuentra una amplia explicación de este comando y otros usados para la administración de procesos.

Este artículo seguirá creciendo de vez en cuando con nueva información sobre comandos que regresan datos valiosos del sistema, asi que chécalo de tiempo en tiempo.
