

Informazioni varie

12.1	Pacchetto Procinfo	363
12.1.1	Utilizzo di «procinfo»	363
12.1.2	Utilizzo di «lsdev»	364
12.1.3	Utilizzo di «socklist»	365
12.2	Pacchetto Linux PCI utilities	365
12.3	Lsusb	366
12.4	Lshw	367
12.5	Definizione del nome del sistema	368
12.6	Altre caratteristiche identificative del sistema	369
12.7	Numero di serie	369
12.8	LinuxInfo	370

hostid 369 hostname 368 368 lsdev 364 lshw 367 lspci 365 lssub 366 procinfo 363 socklist 365 uname 369 \$HOSTNAME 368

In questo capitolo si descrivono sommariamente alcuni programmi che forniscono delle informazioni varie sul sistema in funzione, non avendo trovato una collocazione più specifica nella trattazione di altri argomenti.

12.1 Pacchetto Procinfo

La quantità di informazioni disponibili all'interno di `/proc/` è enorme; inoltre, con l'evolversi dei kernel cambiano i contenuti dei file virtuali e anche la loro collocazione. A questo proposito sono utili i programmi del pacchetto Procinfo¹ che aiutano ad analizzare tali informazioni per generare dei resoconti e delle statistiche più facili da consultare.

12.1.1 Utilizzo di «procinfo»

Il programma `'procinfo'` è quello che, dal pacchetto omonimo, dà le informazioni più comuni sulla gestione dei processi di un sistema GNU/Linux. I dati vengono visualizzati in forma più o meno tabellare e i campi sono indicati attraverso dei nomi. Il significato di alcuni di questi è descritto nella tabella 12.1.

```
procinfo [opzioni]
```

Tabella 12.1. Alcuni dei nomi utilizzati per descrivere i campi delle tabelle generate da `'procinfo'`.

Nome	Descrizione
Memory:	Utilizzo della memoria.
Bootup:	Data e ora dell'avvio del sistema.
Load average:	Carico medio.
user:	Tempo per i processi avviati dagli utenti.
nice:	Tempo per i processi avviati con un valore nice.
system:	Tempo per i processi avviati dal kernel.
idle:	Tempo non utilizzato.
uptime:	Tempo complessivo di funzionamento.
irq <i>n</i> :	Numero di interruzioni e dispositivo corrispondente.
Modules:	Moduli del kernel installati.
Character Devices:	Elenco dei dispositivi a caratteri.
Block Devices:	Elenco dei dispositivi a blocchi.
File Systems:	Tipi di file system gestibili.

Quando `'procinfo'` viene utilizzato senza argomenti si ottengono le informazioni più importanti che possono essere visualizzate su uno schermo normale, per esempio ciò che viene mostrato di seguito:

```
Linux 2.6.14.3 (root@nanohost) (gcc 4.0.2 ) #1 1CPU [nanohost]
Memory:      Total      Used      Free      Shared    Buffers
Mem:         513468    120976    392492      0        21008
Swap:        5462092      0        5462092
```

```

Bootup: Fri Feb 24 08:15:09 2006 Load average: 0.03 0.01 0.00 1/62 7861

user  : 0:00:35.80  0.3% page in : 21573 disk 1: 105r 0w
nice  : 0:00:00.00  0.0% page out: 0
system: 0:00:38.61  0.3% page act: 10203
IOwait: 0:02:08.19  0.9% page dea: 0
hw irq: 0:00:00.82  0.0% page flt: 1679805
sw irq: 0:00:04.65  0.0% swap in : 0
idle  : 3:44:32.08  98.5% swap out: 0
uptime: 3:48:01.64 context : 535302

irq 0: 3419830 timer      irq 12: 95199 i8042
irq 1: 8465 i8042        irq 14: 2192 ide0
irq 2: 0 cascade [4]    irq 15: 15393 ide1
irq 3: 3                 irq 17: 0 Intel 82801BA-ICH2
irq 4: 3                 irq 18: 81824 eth0
irq 6: 9                 irq 19: 0 uhci_hcd:usb1
irq 8: 5 rtc             irq 23: 0 uhci_hcd:usb2

```

Eventualmente, **'procinfo'** può essere utilizzato per ottenere un'informazione continua (o quasi), come fa il programma **'top'**. In questo senso può essere stabilita una pausa tra un aggiornamento e il successivo. Durante questo funzionamento continuo, si possono utilizzare alcuni comandi interattivi, composti da una lettera singola, il cui significato tende a essere coerente con quello delle opzioni della riga di comando. In modo particolare, il comando **'q'** termina il funzionamento continuo di **'procinfo'**.

Tabella 12.3. Alcune opzioni.

Sintassi	Descrizione
-f	Fa sì che 'procinfo' funzioni in modo continuo, a tutto schermo.
-nn_secondi	Questa opzione implica automaticamente la selezione di -f e serve a stabilire un intervallo tra un aggiornamento e l'altro delle informazioni visualizzate.
-m	Mostra le informazioni sui moduli e sui dispositivi a caratteri e a blocchi, trascurando i dati relativi alla CPU e alla memoria.
-a	Mostra tutte le informazioni disponibili, ma per questo non bastano le dimensioni di uno schermo normale.
-d	Mostra le informazioni normali, cioè quelle sull'utilizzo della CPU, della memoria e delle interruzioni (<i>interrupt</i>), ma riferite a periodi di un secondo. Ciò richiede il funzionamento di 'procinfo' in modo continuo, pertanto questa opzione implica automaticamente l'uso di -f .
-Ffile	Ridirige l'output in un file, che di solito corrisponde al dispositivo di una console virtuale inutilizzata.

12.1.2 Utilizzo di «lsdev»

Il programma **'lsdev'** si limita a mostrare una tabella con informazioni tratte dai file **'/proc/interrupts'**, **'/proc/ioports'** e **'/proc/dma'**. In pratica mostra tutti gli indirizzi relativi all'hardware installato.

```
lsdev
```

Il risultato che si ottiene potrebbe essere simile a quello seguente:

```

Device          DMA  IRQ  I/O Ports
-----
                0  1  2  4  9  12  13  14  15
cascade         4
dma             0080-008f
dma1            0000-001f
dma2            00c0-00df
eth0            ff80-ff9f
fdomain        ffa0-ffaf
fpu            00f0-00ff
ide0            01f0-01f7 03f6-03f6 8000-8007
ide1            0170-0177 0376-0376 8008-800f
keyboard       0060-006f
parport0       0378-037a
pic1           0020-003f

```

```

pic2           00a0-00bf
serial         02f8-02ff 03f8-03ff
timer          0040-005f
vga+           03c0-03df

```

12.1.3 Utilizzo di «socklist»

Il programma **'socklist'** si limita a mostrare una tabella con informazioni tratte dai file **'/proc/net/tcp'**, **'/proc/net/udp'** e **'/proc/net/raw'**, integrandoli con le informazioni relative ai descrittori dei file di ogni processo, ovvero **'/proc/*/fd/*'**.

```
socklist
```

Si tratta di informazioni utili per ciò che riguarda la gestione della rete, tuttavia questo programma viene mostrato qui per completare l'argomento del capitolo. Di seguito viene mostrato un esempio del risultato che si può ottenere con **'socklist'**.

type	port	inode	uid	pid	fd	name
tcp	80	246	0	0	0	
tcp	8080	245	0	0	0	
tcp	25	230	0	0	0	
tcp	2049	215	0	0	0	
tcp	515	205	0	0	0	
tcp	635	195	0	0	0	
tcp	53	169	0	0	0	
tcp	53	167	0	0	0	
tcp	98	156	0	0	0	
tcp	113	155	0	0	0	
tcp	37	153	0	0	0	
tcp	79	152	0	0	0	
tcp	143	151	0	0	0	
tcp	110	150	0	0	0	
tcp	109	149	0	0	0	
tcp	513	146	0	0	0	
tcp	514	145	0	0	0	
tcp	70	144	0	0	0	
tcp	23	143	0	0	0	
tcp	21	142	0	0	0	
tcp	111	106	0	0	0	
udp	2049	212	0	0	0	
udp	635	190	0	0	0	
udp	1024	170	0	0	0	
udp	53	168	0	0	0	
udp	53	166	0	0	0	
udp	37	154	0	0	0	
udp	518	148	0	0	0	
udp	517	147	0	0	0	
udp	514	115	0	0	0	
udp	111	105	0	0	0	
raw	1	0	0	0	0	
raw	6	0	0	0	0	

12.2 Pacchetto Linux PCI utilities

Il pacchetto Linux PCI utilities contiene in particolare il programma **'lspci'**² con il quale è possibile conoscere tutte le informazioni disponibili sul bus PCI:

```
lspci [opzioni]
```

Attraverso le opzioni è possibile avere una rappresentazione diversa delle informazioni, ma nella maggior parte dei casi è sufficiente l'utilizzo di **'lspci'** senza argomenti:

```
$ lspci [Invio]
```

```
00:00.0 Host bridge: VIA Technologies, Inc. VT82C693A/694x
↳[Apollo PRO133x] (rev 44)
00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x
↳[Apollo MVP3/Pro133x AGP]
00:07.0 ISA bridge: VIA Technologies, Inc. VT82C596 ISA [Mobile South] (rev 23)
00:07.1 IDE interface: VIA Technologies, Inc.
↳VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC Bus Master IDE (rev 10)
00:07.2 USB Controller: VIA Technologies, Inc. USB (rev 11)
00:07.3 Host bridge: VIA Technologies, Inc. VT82C596 Power Management (rev 30)
00:11.0 Ethernet controller: VIA Technologies, Inc. VT6102 [Rhine-II] (rev 42)
00:12.0 VGA compatible controller: S3 Inc. ViRGE/DX or /GX (rev 01)
00:13.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 08)
00:13.1 Input device controller: Creative Labs SB Live! MIDI/Game Port (rev 08)
00:14.0 SCSI storage controller: Adaptec AIC-7861 (rev 01)
```

Eventualmente basta l'opzione '-v' per ottenere maggiori informazioni:

```
$ lspci -v [Invio]
```

```
00:00.0 Host bridge: VIA Technologies, Inc. VT82C693A/694x
↳[Apollo PRO133x] (rev 44)
    Flags: bus master, medium devsel, latency 0
    Memory at d0000000 (32-bit, prefetchable) [size=64M]
    Capabilities: <available only to root>

00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x
↳[Apollo MVP3/Pro133x AGP] (prog-if 00 [Normal decode])
    Flags: bus master, 66MHz, medium devsel, latency 0
    Bus: primary=00, secondary=01, subordinate=01, sec-latency=0
    Capabilities: <available only to root>

00:07.0 ISA bridge: VIA Technologies, Inc. VT82C596 ISA [Mobile South] (rev 23)
    Subsystem: VIA Technologies, Inc. VT82C596/A/B PCI to ISA Bridge
    Flags: bus master, stepping, medium devsel, latency 0

00:07.1 IDE interface: VIA Technologies, Inc.
↳VT82C586A/B/VT82C686/A/B/VT8233/A/C/VT8235 PIPC Bus Master IDE
↳(rev 10) (prog-if 8a [Master SecP PriP])
    Flags: bus master, medium devsel, latency 32
    I/O ports at d000 [size=16]
    Capabilities: <available only to root>

00:07.2 USB Controller: VIA Technologies, Inc. USB (rev 11) (prog-if 00 [UHCI])
    Subsystem: VIA Technologies, Inc. (Wrong ID) USB Controller
    Flags: bus master, medium devsel, latency 32, IRQ 11
    I/O ports at d400 [size=32]
    Capabilities: <available only to root>

00:07.3 Host bridge: VIA Technologies, Inc. VT82C596 Power Management (rev 30)
    Flags: medium devsel

00:11.0 Ethernet controller: VIA Technologies, Inc. VT6102 [Rhine-II] (rev 42)
    Subsystem: D-Link System Inc DFE-530TX rev A
    Flags: bus master, medium devsel, latency 32, IRQ 11
    I/O ports at d800 [size=256]
    Memory at d9000000 (32-bit, non-prefetchable) [size=256]
    Expansion ROM at <unassigned> [disabled] [size=64K]
    Capabilities: <available only to root>

00:12.0 VGA compatible controller: S3 Inc. ViRGE/DX or /GX
↳(rev 01) (prog-if 00 [VGA])
    Subsystem: S3 Inc. ViRGE/DX
    Flags: bus master, medium devsel, latency 32, IRQ 9
    Memory at d4000000 (32-bit, non-prefetchable) [size=64M]
    Expansion ROM at <unassigned> [disabled] [size=64K]

00:13.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 08)
    Subsystem: Creative Labs CT4832 SBLive! Value
    Flags: bus master, medium devsel, latency 32, IRQ 5
    I/O ports at dc00 [size=32]
    Capabilities: <available only to root>

00:13.1 Input device controller: Creative Labs SB Live! MIDI/Game Port (rev 08)
    Subsystem: Creative Labs Gameport Joystick
    Flags: bus master, medium devsel, latency 32
    I/O ports at e000 [size=8]
    Capabilities: <available only to root>

00:14.0 SCSI storage controller: Adaptec AIC-7861 (rev 01)
    Flags: bus master, medium devsel, latency 32, IRQ 10
    I/O ports at e400 [disabled] [size=256]
    Memory at d9001000 (32-bit, non-prefetchable) [size=4K]
    Expansion ROM at <unassigned> [disabled] [size=64K]
```

12.3 Lsusb

Lsusb³ è un programma che fa parte del pacchetto USBUtils, con lo scopo di elencare i dispositivi USB presenti. Si può usare con delle opzioni, in mancanza delle quali mostra l'elenco completo dei componenti:

```
lsusb [opzioni]
```

Ecco come potrebbe presentarsi l'elenco che si ottiene da questo programma:

```
$ lsusb [Invio]
```

```
Bus 001 Device 003: ID 067b:2517 Prolific Technology, Inc.
↳Flash Disk Mass Storage Device
Bus 001 Device 002: ID 067b:2515 Prolific Technology, Inc.
↳Flash Disk Embedded Hub
Bus 001 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
```

Tabella 12.9. Alcune opzioni.

Sintassi	Descrizione
-v	Richiede la visualizzazione di maggiori informazioni.
--verbose	
-t	Richiede di mostrare la gerarchia tra i componenti USB in forma di albero.

Viene mostrato un esempio in cui si mette a confronto l'uso normale di Lsusb con l'aggiunta successiva dell'opzione '-t':

```
$ lsusb [Invio]
```

```
Bus 001 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 003: ID 04cb:0128 Fuji Photo Film Co., Ltd
Bus 002 Device 001: ID 0000:0000
```

```
$ lsusb -t [Invio]
```

```
Bus# 3
  -Dev# 1 Vendor 0x0000 Product 0x0000
Bus# 2
  -Dev# 1 Vendor 0x0000 Product 0x0000
  -Dev# 3 Vendor 0x04cb Product 0x0128
Bus# 1
  -Dev# 1 Vendor 0x0000 Product 0x0000
```

Eventualmente, per approfondire l'uso del programma si può leggere la pagina di manuale *lsusb(8)*.

12.4 Lshw

Lshw⁴ è un programma in grado di scandire ed elencare tutti i dispositivi esistenti. Attraverso le opzioni della riga di comando è possibile modificare il modo in cui queste informazioni vengono mostrate; altrimenti non c'è bisogno di specificare alcunché:

```
lshw [opzioni]
```

Ecco come potrebbe presentarsi l'elenco che si ottiene da questo programma se avviato dall'utente 'root':

```
# lshw [Invio]
```

```
nanohost
  description: Computer
  product: VT82C692BX
  vendor: VIA Technologies, Inc.
  capabilities: smbios-2.2 dmi-2.2
*-core
  description: Motherboard
  product: 693A-596B-977EF
  physical id: 0
*-firmware
  description: BIOS
  vendor: Award Software International, Inc.
  physical id: 0
  version: 6.00 PG (09/27/2000)
  size: 128KB
  capacity: 192KB
  capabilities: isa pci pnp apm upgrade shadowing escd cdboot
↳bootselect socketedrom eed int13floppy360 int13floppy1200 int13floppy720
↳int13floppy2880 int5sprintscreens int5keyboard int14serial
↳int17printer int10video acpi usb agp ls120boot zipboot biosbootspecification
*-cpu
  description: CPU
  product: Celeron (Coppermine)
  vendor: Intel Corp.
  physical id: 4
  version: 6.8.6
  slot: Socket 370
  size: 766MHz
  clock: 66MHz
  capabilities: fpu fpu_exception wp vme de pae tsc msr pae
↳mce cx8 sep mtrr pge mca cmov pat pse36 mmx fxsr sse
...
...
*-network:5
  description: Ethernet controller
  physical id: 6
```

```

logical name: eth0
serial: 00:50:ba:71:d9:c1
capabilities: mii autonegotiation 100bt-fd 100bt 10bt-fd 10bt ethernet
configuration: autonegotiated=100bt broadcast=yes ←
←driver=via-rhine driverversion=1.1.20-2.6 duplex=full ←
←ip=172.21.1.1 link=yes multicast=yes

```

Si osservi che la precisione e il dettaglio delle informazioni si riduce se usato da un utente comune, senza privilegi.

Tabella 12.13. Alcune opzioni.

Sintassi	Descrizione
<code>-businfo</code>	Mostra le informazioni, in forma tabellare, partendo dalle coordinate del bus.
<code>-C nome_classe</code>	Mostra le informazioni riferite solo alla classe indicata. I nomi stabiliti per la classe sono quelli che si vedono, per esempio, utilizzando l'opzione <code>'-businfo'</code> .
<code>-class nome_classe</code>	

Segue la descrizione di alcuni esempi.

```

• $ lshw -businfo [Invio]

Bus info    Device    Class    Description
-----
                system    Computer
                bus      Motherboard
                memory   System memory
cpu@0
processor   AMD Sempron(tm) 2800+
memory     L1 cache
memory     L2 cache
pci@00:00.0 bridge    746 Host
pci@00:01.0 bridge    SG86C202
...
ide@0      ide0     bus      IDE Channel 0
ide@0.0    /dev/hda disk    Maxtor 6Y120P0
...
usb@2      usb2     bus      OHCI Host Controller
pci@00:03.2 bus      USB 2.0 Controller
usb@1      usb1     bus      EHCI Host Controller
pci@00:04.0 eth0     network  SIS900 PCI Fast Ethernet
...

• $ lshw -businfo -class network [Invio]

Bus info    Device    Class    Description
-----
pci@00:04.0 eth0     network  SIS900 PCI Fast Ethernet
                plip0    network  Ethernet interface
                dummy0   network  Ethernet interface

```

Per maggiori informazioni sull'uso del programma conviene leggere la pagina di manuale *lshw(8)*.

12.5 Definizione del nome del sistema

Nella tradizione dei sistemi Unix, il sistema deve avere un nome. Questo nome tende a confondersi con quello attribuito all'indirizzo dell'interfaccia di rete, se questa esiste; per la precisione si tratta della prima parte, senza il dominio della rete a cui si connette. Tra le altre cose, questo fatto è poi anche motivo di confusione, nel momento in cui si comprende che ci possono essere diverse interfacce di rete, oppure ci possono essere interfacce dinamiche come quelle riferite alle connessioni PPP.

Per un principiante, questa premessa può risultare incomprensibile. In effetti, la gestione della rete viene affrontata in un altro volume, però rimane il fatto che il nome del sistema si attribuisce indipendentemente dalla connessione o meno a una rete, senza nemmeno che ci debba essere necessariamente un'armonizzazione tra questo nome e i nomi utilizzati nell'ambito della rete.

Il nome del sistema si attribuisce con il comando `'hostname'` e generalmente si annota anche all'interno della variabile di ambiente `HOSTNAME`. L'utilizzo di `'hostname'`⁵ è molto semplice:

```
hostname [ nome ]
```

In pratica, se non si indicano argomenti, si ottiene l'emissione del nome attuale; al contrario, se si indica un argomento, quello viene memorizzato come il nome del sistema.

Come si può intuire, la lettura del nome è accessibile a tutti gli utenti, mentre l'impostazione del nome è consentita solo all'utente `'root'`.

In generale, l'impostazione di questa definizione è compito della procedura di inizializzazione del sistema, con la quale si dovrebbe definire coerentemente anche la variabile di ambiente `HOSTNAME`, in modo che contenga lo stesso nome.

In alcuni sistemi, si utilizza il file `'/etc/hostname'` per annotare questo nome, in modo che venga poi letto e utilizzato per la configurazione all'atto dell'avvio del sistema stesso.

12.6 Altre caratteristiche identificative del sistema

Oltre al nome, un sistema dispone anche di altre informazioni identificative. In particolare si tratta del tipo di architettura hardware, il nome del kernel e la sua versione. Queste informazioni si leggono generalmente attraverso il programma `'uname'`:⁶

```
uname [ opzioni ]
```

Il programma `'uname'`, usato senza argomenti, fornisce il nome del kernel, mentre con altri argomenti si possono ottenere informazioni differenti. La tabella 12.16 riepiloga brevemente le opzioni relative.

Tabella 12.16. Opzioni di `'uname'`.

Opzione	Descrizione
<code>-a</code>	Mostra tutte le informazioni disponibili.
<code>--all</code>	
<code>-m</code>	Mostra il nome dell'architettura.
<code>--machine</code>	
<code>-n</code>	Mostra il nome dell'elaboratore.
<code>--nodename</code>	
<code>-o</code>	Mostra il nome del sistema operativo.
<code>--operating-system</code>	
<code>-r</code>	Mostra la versione del sistema operativo.
<code>--kernel-release</code>	
<code>-v</code>	Mostra l'edizione del kernel, nell'ambito della versione riportata dall'opzione <code>'-r'</code> .
<code>--kernel-version</code>	
<code>-s</code>	Mostra il nome del kernel.
<code>--kernel-name</code>	

Si osservi che l'opzione `'-n'` serve a visualizzare lo stesso nome che si ottiene e si imposta con `'hostname'`.

12.7 Numero di serie

Teoricamente, attraverso il programma `'hostid'`⁷ dovrebbe essere possibile ottenere un numero univoco di identificazione dell'elaboratore, determinandolo in base ai componenti fisici che questo contiene (di solito il BIOS):

```
hostid
```

A parte le opzioni minime delle estensioni GNU (`'--help'` e `'--version'`), questo programma non prevede nulla di standard.

Purtroppo, è molto probabile che a elaboratori identici venga attribuito lo stesso numero, rendendo questo programma perfettamente inutile:

```
$ hostid[Invio]
```

```
7F0100
```

12.8 LinuxInfo

◀

LinuxInfo ⁸ è un programma che si comporta in modo simile a `'uname'`, fornendo però informazioni specifiche di un sistema GNU/Linux:

```
linuxinfo [opzioni]
```

Utilizzando `'linuxinfo'` senza argomenti, si ottiene un risultato simile a quello dell'esempio seguente:

```
$ linuxinfo[Invio]
```

```
Linux nanohost 2.6.7 #1 Fri Aug 13 18:28:48 CEST 2004  
One Intel Unknown 768MHz processor, 1511.42 total ↔  
↳bogomips, 384M RAM  
System library 2.3.2
```

¹ **Procnfo** GNU GPL

² **Linux PCI utilities** GNU GPL

³ **USButils** GNU GPL

⁴ **Lshw** GNU GPL

⁵ **GNU core utilities** GNU GPL

⁶ **GNU core utilities** GNU GPL

⁷ **GNU core utilities** GNU GPL

⁸ **LinuxInfo** GNU GPL e GNU LGPL