

1 Montaxe de PCs



Este artigo está en construción. Os autores do mesmo están traballando nel.

Se queres axudar á súa realización ou, simplemente, queres facer algún tipo de comentario, envía un mail a un dos autores que aparecen no pé deste artigo."

1.1 Sumario

- 1 [Introdución](#)
- 2 [Exemplos de montaxes](#)
 - ♦ 2.1 [Equipo básico do ano 2006](#)
 - ♦ 2.2 [Terminal Punto de Venda do ano 2008](#)
- 3 [Aplicacións de Detección de *hardware*](#)
 - ♦ 3.1 [Aplicación *lspci* para linux](#)
 - ♦ 3.2 [Aplicación *dmesg* para linux](#)
 - ♦ 3.3 [Supervisar un sistema linux en funcionamento](#)
- 4 [Aplicacións de Testeo de Equipos e *Bechmarking*](#)
 - ♦ 4.1 [Ciusbet](#)
 - ♦ 4.2 [Phoronix Text Suite](#)

1.2 Introdución

Este apartado dos manuais está dedicado a ver exemplos de montaxe de diversos equipos.

Unha vez ensamblado o equipo non está de máis facer un testeo do equipo, comprobando que non existan ningún tipo de incompatibilidades entre os compoñentes.

1.3 Exemplos de montaxes

1.3.1 Equipo básico do ano 2006

- Elementos **hardware** do equipo a montar:

◊ ...

- [Presentación da Montaxe](#)

1.3.2 Terminal Punto de Venda do ano 2008

- Elementos **hardware** do equipo a montar:

◊ ...

- Pasos da montaxe.

1.4 Aplicacións de Detección de *hardware*

1.4.1 Aplicación *lspci* para linux

Se queremos coñecer os chipset dos elementos conectados ó bus PCI do noso equipo para así buscar e instalar os drivers axeitados en linux, esta é a ferramenta a empregar.

Para obter un resumen de todos os dispositivos que hai conectados ó bus PCI:

```
administrador@ubuntuESI: ~  
Archivo Editar Ver Terminal Solapas Ayuda  
administrador@ubuntuESI:~$ lspci  
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (AGP disabled) (rev 03)  
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 01)  
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)  
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 02)  
00:08.0 VGA compatible controller: S3 Inc. 86c764/765 [Trio32/64/64V+]  
00:0a.0 Ethernet controller: Digital Equipment Corporation DECchip 21140 [FasterNet] (rev 20)
```

Pódense empregar os indicadores **-v** e **-vv** para obter mais información.

A utilidade **lspci** recupera do bus PCI algunha información, mostrando datos adicionais obtidos da súa propia base de datos almacenada en **/usr/share/misc/pci.ids**. Tamén existe un comando para actualizar ese arquivo: **# update-pciids**

```
root@ubuntuESI:/# update-pciids  
--19:53:45-- http://pciids.sourceforge.net/v2.2/pci.ids.bz2  
=> '/usr/share/misc/pci.ids.gz.new'  
Resolviendo pciids.sourceforge.net... 66.35.250.209  
Conectando a pciids.sourceforge.net[66.35.250.209]:80... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 132,419 (129K) [text/plain]  
  
100%[=====>] 132,419 11.09K/s ETA 00:00  
  
19:54:17 (8.14 KB/s) - '/usr/share/misc/pci.ids.gz.new' guardado [132419/132419]  
  
Done.  
root@ubuntuESI:/#
```

- [Páxina man de lspci](#)

1.4.2 Aplicación *dmesg* para linux

Para obter unha lista de todos os compoñentes detectados polo núcleo pódese empregar o comando **dmesg**. Para ver todo o contido deste registro procederemos do seguinte xeito:

```
$ dmesg | less
```

Tamén se pode filtrar o resultado de **dmesg** para atopar dispositivos específicos. Vexamos un par de exemplos:

- Obter a memoria RAM instalada no equipo faríase así:

```
$ dmesg | grep -i memory  
Memory: 1031480k/1047424k available (2250k kernel code, 15144k reserved, 1202k data, 268k init, 129920k highmem)  
virtual kernel memory layout:  
Freeing initrd memory: 1506k freed  
Total HugeTLB memory allocated, 0  
Non-volatile memory driver v1.2  
Freeing unused kernel memory: 268k freed  
[drm] Setting GART location based on new memory map
```

Como se ve o equipo ten instalado 1 GB de RAM.

- Comprobar cal é o Microprocesador instalado:

```
$ dmesg | grep -i cpu  
Initializing CPU#0  
CPU 0 irqstacks, hard=c07cd000 soft=c07ad000  
SLUB: Genslabs=22, HWalign=64, Order=0-1, MinObjects=4, CPUs=1, Nodes=1
```

```
CPU: After generic identify, caps: 078bfbff e3d3fbff 00000000 00000000 00000001 00000000 00000001
CPU: L1 I Cache: 64K (64 bytes/line), D cache 64K (64 bytes/line)
CPU: L2 Cache: 512K (64 bytes/line)
CPU: After all inits, caps: 078bf3ff e3d3fbff 00000000 00000410 00000001 00000000 00000001
Intel machine check reporting enabled on CPU#0.
CPU0: AMD Athlon(tm) 64 Processor 3200+ stepping 02
Brought up 1 CPUs
```

Como vemos, o equipo ten instalado un AMD Athlon 64 3200+.

- [Páxina man de dmesg](#)

1.4.3 Supervisar un sistema linux en funcionamento

Se se quere supervisar un sistema linux en funcionamento e en tempo real, vendo información relativa á memoria física, a CPU, as unidades de disco, etc. pódese empregar o sistema de arquivos */proc*.

Neste caso, simplemente, hai que empregar o comando **cat** para ir lendo os arquivos de texto existentes no seu interior (tamén poderíamos utilizar unha das utilidades deseñadas para tal fin como sysctl, lspci, ps ou top). A sintaxe é a mesma para ler calquera dos arquivos existentes:

```
$ cat /proc/[nome do arquivo a ler]
```

Vexamos un par de exemplos:

•

1.5 Aplicacións de Testeo de Equipos e *Bechmarking*

Neste apartado veremos varias utilidades software que nos permitirán comprobar que os equipos ensamblados non presentan incompatibilidades entre compoñentes. Ademais, a maioría destas utilidades, nos darán información comparativa (*bechmarking*) entre o equipo analizado e outros xenéricos dos que ten información gardada nunha base de datos.

O **Benchmarking** é unha técnica que consiste en analizar o rendemento do *hardware* xeralmente comparado con algún parámetro como pode ser o de algún sistema en concreto ou un valor arbitrario. Esta técnica, pode parecer pouco relevante, ou simplemente curiosa. Pero a verdade é que o *benchmarking* é fundamental, tanto no deseño de *hardware* como á hora de decidir que compoñente é o que precisamos.

1.5.1 Ciusbet



Ciusbet analiza de forma eficiente o rendemento dos distintos compoñentes do ordenador con *software* de *benchmarking* propietario.

- [Web principal de Ciusbet](#)

Na seguinte imaxe vemos o resultado de executar o software **Ciusbet CPU Benchmark** nun equipo Dell Latitude D430 con micro Intel Core2 U7500 a 1,06GHz e 2 GB de RAM.

```
C:\WINDOWS\system32\cmd.exe - CHB_MT.exe

-----
|          TEST          |      TIEMPO      |  RENDIMIENTO  |  THREADS  |  SCORE  |
-----
CPU Logical Test         13656 ms         105.15%       4          1050 PTOS
CPU Conditional Jump     16485 ms         143.79%       4          1440 PTOS
CPU Integer Arithmetic I  47500 ms         202.34%       4          2020 PTOS
CPU Integer Arithmetic II 124328 ms         54.54%       4           550 PTOS
CPU BruteForce           12578 ms         154.16%       4           770 PTOS
CPU Array Arithmetics    22391 ms         111.93%       4           560 PTOS
CPU Float FPU I          43922 ms          19.35%       4           290 PTOS
CPU Float FPU II         9062 ms          179.14%       4          2690 PTOS
CPU Cache Speed          11985 ms         126.07%       1          2520 PTOS
-----
TOTAL:                                     1190 PTOS

-----<Giusbet CPU BenchMark: Resultado>-----
!
!Modelo CPU:  Intel(R) Core(TM)2 CPU          U7500 @ 1.06GHz
!Num Cores :   2
!U. Reloj  :   1063 MHz
!Puntuacion:  1190 puntos
-----

Tu CPU
Intel Core Duo T2050 <1,60 GHz> : #####
Intel Pentium D      <3,40 GHz> : #####
AMD Athlon 64 3200+  <2,00 GHz> : #####
AMD Athlon XP 2400+  <2,00 GHz> : #####
Intel Pentium 4       <2,40 GHz> : #####
```

1.5.2 Phoronix Text Suite

Phoronix Test Suite (PTS) é unha ferramenta, baixo GPL, para realizar diversos *benchmarks* de rendemento de *hardware* para GNU/Linux (aínda que podería chegar a funcionar noutros sistemas).

Con PTS os desenvolvedores de aplicacións poderán construír facilmente os *benchmarks* do seu propio *software* e os usuarios poderán medir facilmente o rendemento dos seus equipos en devandito *software*.

Outro aspecto interesante é a posibilidade de subir os resultados ao servizo PTS Global e poder executar, introducindo o código dunha proba xa existente, calquera *benchmark* que estea dispoñible en PTS Global.

A versión actual de PTS é a 0.1. Por agora non ten interface gráfica e as probas hai que executalas desde un terminal, a pesar disto o uso é bastante sinxelo.

- [Páxina web do proxecto Phoronix](#)