

lspci

- Objet : la commande lspci
- Niveau requis :
[débutant, avisé](#)
- Commentaires : *Pour lister l'ensemble des périphériques connectés aux bus PCI du système*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
 - Création par [mattux](#) le 17/06/2007
 - Testé par [milou](#) le 31/07/2015
 - Modification par [bendia](#) le 6/11/2015
- Commentaires sur le forum : [C'est ici](#)¹⁾

Voir : [ls](#)

La commande pour les périphériques USB : [lsusb](#)

Introduction

La commande `lspci` permet de connaître la liste du matériel (cartes, *chipsets*, contrôleurs, etc.) de votre PC utilisant l'interface PCI.

Syntaxe

```
lspci
```

Donne par exemple :

[retour de la commande](#)

```
00:00.0 Host bridge: Advanced Micro Devices, Inc. [AMD] RS880 Host
Bridge
00:01.0 PCI bridge: ASUSTeK Computer Inc. AMD RS780/RS880 PCI to PCI
bridge (int gfx)
00:09.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] RS780/RS880 PCI
to PCI bridge (PCIE port 4)
00:0a.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] RS780/RS880 PCI
to PCI bridge (PCIE port 5)
00:11.0 SATA controller: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 SATA Controller [IDE mode]
00:12.0 USB controller: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB OHCI0 Controller
00:12.1 USB controller: Advanced Micro Devices, Inc. [AMD/ATI] SB7x0
USB OHCI1 Controller
00:12.2 USB controller: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB EHCI Controller
```

```
00:13.0 USB controller: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB OHCI0 Controller
00:13.1 USB controller: Advanced Micro Devices, Inc. [AMD/ATI] SB7x0
USB OHCI1 Controller
00:13.2 USB controller: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB EHCI Controller
00:14.0 SMBus: Advanced Micro Devices, Inc. [AMD/ATI] SBx00 SMBus
Controller (rev 3c)
00:14.1 IDE interface: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 IDE Controller
00:14.2 Audio device: Advanced Micro Devices, Inc. [AMD/ATI] SBx00
Azalia (Intel HDA)
00:14.3 ISA bridge: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 LPC host controller
00:14.4 PCI bridge: Advanced Micro Devices, Inc. [AMD/ATI] SBx00 PCI to
PCI Bridge
00:14.5 USB controller: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB OHCI2 Controller
00:18.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Family 10h
Processor HyperTransport Configuration
00:18.1 Host bridge: Advanced Micro Devices, Inc. [AMD] Family 10h
Processor Address Map
00:18.2 Host bridge: Advanced Micro Devices, Inc. [AMD] Family 10h
Processor DRAM Controller
00:18.3 Host bridge: Advanced Micro Devices, Inc. [AMD] Family 10h
Processor Miscellaneous Control
00:18.4 Host bridge: Advanced Micro Devices, Inc. [AMD] Family 10h
Processor Link Control
01:05.0 VGA compatible controller: Advanced Micro Devices, Inc.
[AMD/ATI] RS880 [Radeon HD 4250]
01:05.1 Audio device: Advanced Micro Devices, Inc. [AMD/ATI] RS880 HDMI
Audio [Radeon HD 4200 Series]
02:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd.
RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 06)
03:00.0 USB controller: NEC Corporation uPD720200 USB 3.0 Host
Controller (rev 03)
04:05.0 Multimedia video controller: Conexant Systems, Inc.
CX23880/1/2/3 PCI Video and Audio Decoder (rev 05)
04:05.1 Multimedia controller: Conexant Systems, Inc. CX23880/1/2/3 PCI
Video and Audio Decoder [Audio Port] (rev 05)
04:05.2 Multimedia controller: Conexant Systems, Inc. CX23880/1/2/3 PCI
Video and Audio Decoder [MPEG Port] (rev 05)
04:08.0 FireWire (IEEE 1394): VIA Technologies, Inc. VT6306/7/8 [Fire
II(M)] IEEE 1394 OHCI Controller (rev c0)
```

Le retour de cette commande peut se décomposer comme suit :

```
<Numéro de bus>:<Numéro de périphérique>.<Fonction> <Classe du
périphérique>: <Nom du constructeur> <Modèle du périphérique> (<Révision du
périphérique>)
```

Magique non ? Comment le système fait-il pour obtenir automatiquement ces informations ? Hé bien chaque composant du bus PCI contient un certain nombre d'informations directement dans sa mémoire, ce qui permet une auto-configuration du système.

On peut d'ailleurs voir à quoi ressemblent les informations enregistrées directement dans le périphérique avec :

```
lspci -n
```

[retour de la commande](#)

```
00:00.0 0600: 1022:9601
00:01.0 0604: 1043:9602
00:09.0 0604: 1022:9608
00:0a.0 0604: 1022:9609
00:11.0 0106: 1002:4390
00:12.0 0c03: 1002:4397
00:12.1 0c03: 1002:4398
00:12.2 0c03: 1002:4396
00:13.0 0c03: 1002:4397
00:13.1 0c03: 1002:4398
00:13.2 0c03: 1002:4396
00:14.0 0c05: 1002:4385 (rev 3c)
00:14.1 0101: 1002:439c
00:14.2 0403: 1002:4383
00:14.3 0601: 1002:439d
00:14.4 0604: 1002:4384
00:14.5 0c03: 1002:4399
00:18.0 0600: 1022:1200
00:18.1 0600: 1022:1201
00:18.2 0600: 1022:1202
00:18.3 0600: 1022:1203
00:18.4 0600: 1022:1204
01:05.0 0300: 1002:9715
01:05.1 0403: 1002:970f
02:00.0 0200: 10ec:8168 (rev 06)
03:00.0 0c03: 1033:0194 (rev 03)
04:05.0 0400: 14f1:8800 (rev 05)
04:05.1 0480: 14f1:8801 (rev 05)
04:05.2 0480: 14f1:8802 (rev 05)
04:08.0 0c00: 1106:3044 (rev c0)
```

Moins lisible, n'est-ce pas ? C'est cependant ce mode d'affichage qui permet de retrouver avec certitude le modèle d'un périphérique. On peut également afficher à la fois le nom et les identifiants numériques :

```
lspci -nn
```

[retour de la commande](#)

```
00:00.0 Host bridge [0600]: Advanced Micro Devices, Inc. [AMD] RS880
Host Bridge [1022:9601]
00:01.0 PCI bridge [0604]: ASUSTeK Computer Inc. AMD RS780/RS880 PCI to
PCI bridge (int gfx) [1043:9602]
00:09.0 PCI bridge [0604]: Advanced Micro Devices, Inc. [AMD]
RS780/RS880 PCI to PCI bridge (PCIE port 4) [1022:9608]
00:0a.0 PCI bridge [0604]: Advanced Micro Devices, Inc. [AMD]
RS780/RS880 PCI to PCI bridge (PCIE port 5) [1022:9609]
00:11.0 SATA controller [0106]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 SATA Controller [IDE mode] [1002:4390]
00:12.0 USB controller [0c03]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB OHCI0 Controller [1002:4397]
00:12.1 USB controller [0c03]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0 USB OHCI1 Controller [1002:4398]
00:12.2 USB controller [0c03]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB EHCI Controller [1002:4396]
00:13.0 USB controller [0c03]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB OHCI0 Controller [1002:4397]
00:13.1 USB controller [0c03]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0 USB OHCI1 Controller [1002:4398]
00:13.2 USB controller [0c03]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB EHCI Controller [1002:4396]
00:14.0 SMBus [0c05]: Advanced Micro Devices, Inc. [AMD/ATI] SBx00
SMBus Controller [1002:4385] (rev 3c)
00:14.1 IDE interface [0101]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 IDE Controller [1002:439c]
00:14.2 Audio device [0403]: Advanced Micro Devices, Inc. [AMD/ATI]
SBx00 Azalia (Intel HDA) [1002:4383]
00:14.3 ISA bridge [0601]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 LPC host controller [1002:439d]
00:14.4 PCI bridge [0604]: Advanced Micro Devices, Inc. [AMD/ATI] SBx00
PCI to PCI Bridge [1002:4384]
00:14.5 USB controller [0c03]: Advanced Micro Devices, Inc. [AMD/ATI]
SB7x0/SB8x0/SB9x0 USB OHCI2 Controller [1002:4399]
00:18.0 Host bridge [0600]: Advanced Micro Devices, Inc. [AMD] Family
10h Processor HyperTransport Configuration [1022:1200]
00:18.1 Host bridge [0600]: Advanced Micro Devices, Inc. [AMD] Family
10h Processor Address Map [1022:1201]
00:18.2 Host bridge [0600]: Advanced Micro Devices, Inc. [AMD] Family
10h Processor DRAM Controller [1022:1202]
00:18.3 Host bridge [0600]: Advanced Micro Devices, Inc. [AMD] Family
10h Processor Miscellaneous Control [1022:1203]
00:18.4 Host bridge [0600]: Advanced Micro Devices, Inc. [AMD] Family
10h Processor Link Control [1022:1204]
01:05.0 VGA compatible controller [0300]: Advanced Micro Devices, Inc.
[AMD/ATI] RS880 [Radeon HD 4250] [1002:9715]
01:05.1 Audio device [0403]: Advanced Micro Devices, Inc. [AMD/ATI]
RS880 HDMI Audio [Radeon HD 4200 Series] [1002:970f]
02:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd.
RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller [10ec:8168]
(rev 06)
```

```
03:00.0 USB controller [0c03]: NEC Corporation uPD720200 USB 3.0 Host
Controller [1033:0194] (rev 03)
04:05.0 Multimedia video controller [0400]: Conexant Systems, Inc.
CX23880/1/2/3 PCI Video and Audio Decoder [14f1:8800] (rev 05)
04:05.1 Multimedia controller [0480]: Conexant Systems, Inc.
CX23880/1/2/3 PCI Video and Audio Decoder [Audio Port] [14f1:8801] (rev
05)
04:05.2 Multimedia controller [0480]: Conexant Systems, Inc.
CX23880/1/2/3 PCI Video and Audio Decoder [MPEG Port] [14f1:8802] (rev
05)
04:08.0 FireWire (IEEE 1394) [0c00]: VIA Technologies, Inc. VT6306/7/8
[Fire II(M)] IEEE 1394 OHCI Controller [1106:3044] (rev c0)
```

Options de tri

Bon, c'est bien joli tout ça, mais on se rend compte que c'est un peu long, comme retour. `lspci` renvoie en effet une ligne par périphérique... lorsqu'on l'utilise avec son mode le moins bavard. Voyons voir s'il est possible de filtrer ce résultat.

Filtrer par slot (dit "emplacement")

On peut filtrer le résultat par *slot*, c'est-à-dire le code donné dans le premier champ grâce à l'option `-s`. Exemple :

```
lspci -s 04:08.0
```

[retour de la commande](#)

```
4:08.0 FireWire (IEEE 1394): VIA Technologies, Inc. VT6306/7/8 [Fire
II(M)] IEEE 1394 OHCI Controller (rev c0)
```

Filtrer par nom de périphérique

Il est également possible de filtrer grâce au nom du périphérique avec l'option `-d`. Pas directement son nom en réalité, mais grâce au code `IdentifiantVendeur:IdentifiantPériphérique`. Exemple :

```
lspci -d 1106:3044
```

[retour de la commande](#)

```
04:08.0 FireWire (IEEE 1394): VIA Technologies, Inc. VT6306/7/8 [Fire
II(M)] IEEE 1394 OHCI Controller (rev c0)
```

Filtrer par classe de périphériques

Il est également possible de filtrer par classe de périphériques²⁾, toujours avec l'option -d. Exemple :

```
lspci -d ::0480
```

[retour de la commande](#)

```
04:05.1 Multimedia controller: Conexant Systems, Inc. CX23880/1/2/3 PCI
Video and Audio Decoder [Audio Port] (rev 05)
04:05.2 Multimedia controller: Conexant Systems, Inc. CX23880/1/2/3 PCI
Video and Audio Decoder [MPEG Port] (rev 05)
```

Mais pourquoi filtrer ? Parce que la commande peut être nettement plus **bavarde** et renvoyer plusieurs lignes par périphérique, au point qu'il est difficile de s'y retrouver.

Options d'affichage

Les options -v, -vv et -vvv permettent d'augmenter la quantité d'informations affichées. Exemple :

```
lspci -vd 1106:3044
```

[retour de la commande](#)

```
04:08.0 FireWire (IEEE 1394): VIA Technologies, Inc. VT6306/7/8 [Fire
II(M)] IEEE 1394 OHCI Controller (rev c0) (prog-if 10 [OHCI])
  Subsystem: ASUSTeK Computer Inc. Motherboard
  Flags: bus master, medium devsel, latency 64, IRQ 23
  Memory at febff800 (32-bit, non-prefetchable) [size=2K]
  I/O ports at ec00 [size=128]
  Capabilities: [50] Power Management version 2
  Kernel driver in use: firewire_ohci
```

On voit apparaître ici une information fort intéressante, à savoir le pilote (ze *driver*, en anglais 🤖) utilisé : `Kernel driver in use`. Il est possible de n'ajouter que cette information (ou presque) à la sortie habituelle avec l'option -k. Exemple :

```
lspci -kd 1106:3044
```

[retour de la commande](#)

```
04:08.0 FireWire (IEEE 1394): VIA Technologies, Inc. VT6306/7/8 [Fire
II(M)] IEEE 1394 OHCI Controller (rev c0)
  Subsystem: ASUSTeK Computer Inc. Motherboard
```

```
Kernel driver in use: firewire_ohci
```

Périphérique "inconnu"

Vous venez d'acheter la dernière carte graphique méga géniale qui affiche 30 millions de *shadow* et en plus elle fait le café, mais voilà, `lspci` vous retourne un déprimant Device... Votre carte est inconnue au bataillon.

Souvenez-vous, nous avons dit tout au début que les informations permettant d'identifier le matériel se trouvent directement dans la carte sous forme de code cabalistique (hexadécimal, en réalité 😊). Pour afficher des informations lisibles par le débianeux moyen, `lspci` fait correspondre les codes hexadécimaux aux données présentes dans le fichier `/usr/share/misc/pci.ids`. On peut mettre à jour ce fichier avec la commande :

```
update-pciids
```

[retour de la commande](#)

```
Downloaded daily snapshot dated 2018-12-18 03:15:01
```

Et puisque le débianeux moyen est vachement curieux (ou curieuse, il paraît que le neutre se conjugue au masculin 😊) :

```
ls -lh /usr/share/misc/pci.ids*
```

[retour de la commande](#)

```
-rw-r--r-- 1 root root 1,1M déc. 18 03:15 /usr/share/misc/pci.ids
-rw-r--r-- 1 root root 1,1M nov. 30 2016 /usr/share/misc/pci.ids.old
```

Pense-bête



- tableau résumant les options

Liens utiles

- Manuel de l'administrateur Linux (2008) : <https://delafond.org/traducmanfr/man/man8/lspci.8.html>
- (en) *The PCI ID Repository* (dépôt de référence pour les identifiants PCI) : <https://pci-ids.ucw.cz/>
- (en) *Decoding PCI data and lspci output on Linux hosts* :

<https://prefetch.net/articles/linuxpci.html>

¹⁾

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

²⁾

À partir de la version 3.3.0 des *PCI Utilities*, soit en pratique à partir de Debian 9 Stretch.

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:systeme:lspci>



Last update: **30/11/2020 19:08**