

QUELLE EST MA CONFIGURATION MATÉRIELLE ?

par Fleur Brosseau

Que contient mon ordinateur ?

Votre ordinateur, quel qu'il soit, comporte au moins l'ensemble des éléments suivants :

■ **une carte mère** : l'élément de base de votre ordinateur. C'est un grand circuit imprimé qui permet de relier entre eux tous les éléments essentiels d'un ordinateur. Elle possède des connecteurs pour les cartes, les barrettes de mémoire, le processeur, etc. Chaque carte mère possède un *chipset*, un jeu de composants électroniques permettant de faire le lien entre le ou les processeur(s), la mémoire et les périphériques.

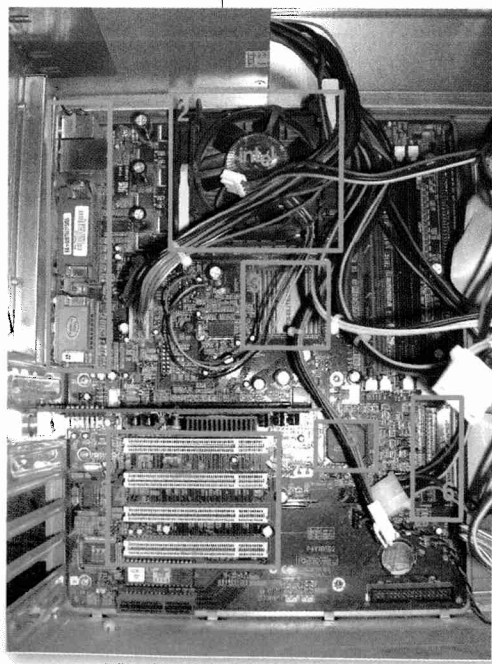
■ **un processeur** : aussi appelé « CPU » (pour *Central Processing Unit*), est le composant essentiel de l'ordinateur. C'est lui qui exécute les programmes. Un processeur est caractérisé par sa marque, son modèle et sa vitesse.

■ **une carte graphique** : (aussi appelée « carte vidéo ») parfois intégrée directement dans la carte mère. Cette carte est chargée de convertir les données numériques en éléments graphiques pour un affichage à l'écran.

■ **de la mémoire vive** : appelée « mémoire vive » ou « RAM » (*Random Access Memory*), permet de stocker des données de manière temporaire lors de l'exécution d'un programme, d'une application. Ce type de mémoire est volatile : lorsque vous éteignez l'ordinateur, la mémoire est vidée.

■ **un disque dur** : qui vous permettra de stocker toutes vos données de manière permanente, contrairement à la mémoire vive (c'est pourquoi on l'appelle également « mémoire de masse »).

Fig. 1 :
- 1 : connecteurs d'entrée/sortie
- 2 : ventilateur (sous lequel se cache le processeur)
- 3 : chipset1
- 4 : connecteurs PCI
- 5 : chipset 2
- 6 : connecteurs pour disques et lecteurs (ports IDE).



Comme vous le savez maintenant, le monde du Libre est peuplé de nombreux forums d'entraide pour les utilisateurs. Mais la règle d'or est toujours la même : accompagner sa question d'une description précise du matériel qui pose problème. En effet, difficile d'aider quelqu'un sans connaître sa configuration matérielle exacte. Il est donc essentiel de connaître sur quoi repose son système, non seulement pour pouvoir définir précisément son environnement de travail en cas de problème, mais aussi par simple curiosité personnelle...

À ces composants peuvent s'ajouter d'autres cartes selon vos besoins (qui sont parfois livrées avec l'ordinateur que vous avez acheté dans le commerce) : cartes son, cartes réseau, carte d'acquisition vidéo, etc.

Ces cartes supplémentaires peuvent soit être intégrées directement à la carte mère, soit se trouver sur des ports d'extension (ou *slot*). Dans tous les cas, elles sont connectées via un bus PCI (*Peripheral Component Interconnect*), celui-ci pouvant recevoir tout type de carte. En moyenne, une carte mère dispose de 3 ou 4 connecteurs PCI, reconnaissables à leur couleur blanche.

De quelles cartes je dispose ?

Pour savoir de quelles cartes supplémentaires vous disposez sur votre ordinateur, une seule commande à retenir : `lspci`. Cet outil fait partie du paquet `pciutils` (à installer donc, si votre système ne connaît pas cette commande). Notez que vous devez être *root* pour pouvoir utiliser la commande `lspci`.

Chaque carte est connectée à un bus PCI et possède une position sur ce bus. De plus, chaque carte est caractérisée par une classe, une marque, un modèle et une révision. La commande `lspci -n` permet de connaître ces informations.

Quel est l'intérêt de ces valeurs numériques ? Eh bien, ces valeurs sont utilisées par un pilote (ou module) pour savoir s'il est compatible avec les cartes présentes sur votre système. Si le driver reconnaît l'identifiant d'une carte, alors il peut la piloter.

La commande `lspci -v` permet de connaître diverses informations supplémentaires comme la fréquence du bus associé à la carte, les zones mémoires qu'elle occupe, etc.

Vous trouverez à la page suivante : <http://kmuto.jp/debian/hcl/> un outil très pratique. Il s'agit d'une page de vérification des pilotes de périphériques pour GNU/Linux. Il suffit d'y coller ce que vous retourne la commande `lspci -n` et vous saurez en un clic quel matériel du bus PCI est pris en charge et par quel pilote.

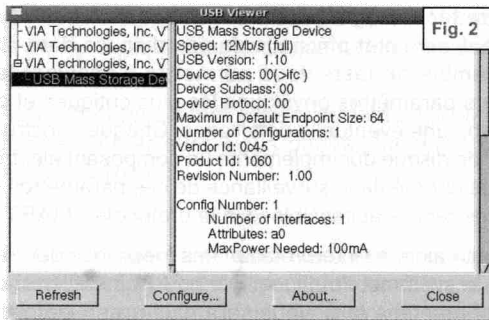


Fig. 2

Enfin, pour connaître la liste du matériel USB connecté à votre machine, il suffit de saisir la commande : `lsusb`. Cet outil fait partie du *package* `usbutils`.

`lsusb` lit les informations concernant les périphériques USB dans le répertoire `/proc/bus/usb`. Vous pouvez visualiser ces informations de manière plus conviviale via une interface graphique comme `Usbviewer` ou encore via le gestionnaire de périphériques que proposent la plupart des environnements de bureau.

Connaître sa carte mère

DMI (pour *Desktop Management Interface*) est un standard défini par la DMTF (*Desktop Management Task Force*), permettant de décrire les ressources matérielles d'un ordinateur (<http://www.dmtf.org/standards/dmi/>). Ce standard permet ainsi d'obtenir plusieurs types d'informations sur votre matériel, mais il est parfois très mal implémenté par les fabricants. Vous n'obtiendrez donc peut-être pas la totalité des informations...

L'outil `dmidecode` permet d'afficher à l'écran ces informations de manière suffisamment intelligible. Cette commande ne fait qu'interpréter les informations telles qu'elles sont décrites dans le BIOS, selon le standard SMBIOS/DMI. Ceci inclut le fabricant, le modèle, le numéro de série, la version du BIOS et bien d'autres renseignements encore selon le fabricant. La spécification SMBIOS permet de définir 39 « types DMI ». Par exemple :

■ Le type 0 contient les informations du BIOS de la carte mère. On y retrouve la marque du constructeur du BIOS, sa version, sa date et sa taille.

■ Le type 2 nous renseigne au sujet de la marque et du modèle de carte mère.

■ Le type 17 renvoie le nombre de slots de mémoires disponibles et leur configuration.

Pour accéder à l'un ou l'autre de ces types, il faut utiliser l'option `-t`, suivie du numéro du type que vous souhaitez visualiser. Par exemple : `dmidecode -t 17` (notons que cette option n'est disponible que pour les versions récentes de `dmidecode`).

Connaître son processeur

La commande `cat /proc/cpuinfo` retourne toutes les informations concernant le (ou les) processeur(s) de votre système.

Vous pouvez ainsi visualiser le nom du fabricant, le modèle, la fréquence, etc.

La commande `top` permet de visualiser l'activité du processeur en temps réel. Plus précisément, il affiche la quantité de mémoire utilisée par processus, ainsi que les pourcentages d'utilisation du processus. Notons, que le moniteur système de l'environnement Gnome (*Applications -> Outils système -> Moniteur système*) permet d'accéder à l'utilisation du processeur en temps réel via une interface graphique. De même, sous KDE, c'est le centre de contrôle de KDE qui vous fournit ces informations, via une interface graphique intuitive.

Mon système Linux

■ Quel est le nom de ma machine ? `$ hostname`

■ Quelle est ma version du noyau ? `$ uname -r`

À noter que la commande `cat /proc/version` retourne également la version du noyau, avec parfois plus de détails.

■ Quels sont les modules chargés par le noyau ? `$ lsmod`

Cette commande interprète en fait le contenu du fichier `/proc/modules`, en affichant les modules du noyau qui sont actuellement chargés.

Récupérer des informations sur les disques IDE

IDE est une interface standard permettant la connexion de périphériques de stockage sur un ordinateur, comme les disques durs, mais aussi lecteurs CD-ROM et DVD. C'est l'outil `hdparm` qui permet d'agir sur les périphériques IDE. En particulier, il permet de consulter les informations se rapportant à un disque, avec l'option `-i` suivie du nom du périphérique que vous souhaitez visualiser.

La première ligne nous renseigne au sujet de la marque, du modèle et du numéro de série du disque dur. Remarquez que la mention *BuffType* renvoie le type de mémoire cache utilisée par le disque dur et la mention *BuffSize* renvoie sa taille. Rappelons qu'une taille de mémoire cache élevée pour un disque dur permet d'en augmenter les performances.

La commande `hdparm -t` permet de tester la vitesse de lecture de votre disque. Ce sont les premiers secteurs du disque qui sont testés (les plus rapides), pendant 3 secondes. La valeur renvoyée n'est pas une moyenne, mais une valeur maximale de la vitesse.

```
$ sudo hdparm -t /dev/hda
/dev/hda:
Timing buffered disk reads: 46 MB in 3.06 seconds =
15.02 MB/sec
```

On peut considérer que la vitesse d'un disque dur est généralement comprise entre 10 et 30 MB/sec.

Toutes les infos en un coup d'œil avec LSHW

Un outil permet de récapituler toutes les informations précédentes et bien d'autres encore. Il s'agit de la commande `lshw` (à lancer en tant que `root`). LSHW signifie *HardWare LiSter*, ce qui est on ne peut plus parlant. Cet outil affiche de façon claire et détaillée, des informations concernant la mémoire, la version du *firmware*, la configuration de la carte mère, la version du CPU et sa vitesse, la configuration du cache, la vitesse du bus, les paramètres réseau, etc.

Par défaut, le résultat est affiché sur la sortie standard. Pour le conserver, il suffit donc de faire une redirection dans un fichier. Le résultat peut être extrait au format texte, XML ou HTML. Ainsi la commande `lshw` peut s'accompagner de l'option `-html` qui permet de consulter le résultat directement sur une page Web. Et pour ceux qui sont plus à l'aide avec une interface graphique, sachez qu'il existe un package `lshw-gtk` qui rend l'affichage plus attractif.

```
# lshw -html > ma_config.html
```

Vous obtenez un fichier `ma_config.html` consultable dans n'importe quel navigateur Web. Les différentes informations sont classées par catégorie, sous forme d'arborescence.

Dans quel état est mon disque dur ?

La perte de données suite à un crash du disque dur est toujours navrant. D'autant plus qu'un disque la plupart du temps donne des signes avant-coureurs de son état de faiblesse. Ce serait tellement plus simple de pouvoir repérer ces signes pour éviter la panne fatale. C'est pour cela que la technologie SMART (*Self-Monitoring, Analysis and Reporting Technology systems*) a vu le jour.

ID#	ATTRIBUTE_NAME	FLAG	VALUE	WORST	THRESH	TYPE	UPDATED	WHEN_FAILED	RAW_VALUE
1	Raw_Read_Error_Rate	0x0006	100	100	016	Pre-fail Always	-	0	
2	Throughput_Performance	0x0005	100	100	050	Pre-fail Offline	-	0	
3	Spin_Up_Time	0x0007	100	100	024	Pre-fail Always	-	200 (Average 200)	
4	Start_Stop_Count	0x0012	100	100	000	Old-age Always	-	264	
5	Reallocated_Sector_Ct	0x0038	100	100	005	Pre-fail Always	-	0	
7	Seek_Error_Rate	0x000b	100	100	067	Pre-fail Always	-	0	
8	Seek_Time_Performance	0x0005	100	100	020	Pre-fail Offline	-	0	
9	Power_On_Hours	0x0012	100	100	000	Old-age Always	-	3814	
10	Spin_Retry_Count	0x0013	100	100	060	Pre-fail Always	-	0	
12	Power_Cycle_Count	0x0032	100	100	000	Old-age Always	-	263	

Fig. 3

Cette technologie permet en effet de connaître et d'analyser l'état physique de vos disques durs. Un ensemble de tests vous permettra d'analyser les divers paramètres physiques les plus critiques et de prévoir une éventuelle défaillance. Chaque constructeur de disque dur implémente un composant électronique chargé de la surveillance de ces paramètres et de les rendre accessibles via le protocole SMART.

Et pour aider à l'interprétation des messages délivrés par le système, l'outil `smartmontools` (à installer si ce n'est pas déjà fait), vous sera d'un grand secours. Très simple à utiliser, `smartmontools` fournit une quantité d'informations : température du disque, taux d'erreurs de lecture/écriture, etc. Il se compose de deux éléments :

- Un démon, `smartd`, analyse périodiquement, toutes les 30 minutes par défaut, les paramètres fournis par le disque dur. On le configure simplement à l'aide du fichier `/etc/smartd.conf`.

- La commande `smartctl` (à utiliser avec les privilèges du super-utilisateur), qui avec les options appropriées suivies de l'identification du périphérique concerné, vous permettra d'effectuer divers tests.

Commençons avec quelque chose de basique, obtenir des informations sur le disque (`/dev/hda` correspond ici au disque dur) :

```
# smartctl -i /dev/hda
```

L'option `-a` retourne, quant à elle, l'état des différents attributs du disque (vous trouverez la définition de ces attributs à la page <http://smartlinux.sourceforge.net/smart/attributes.php>). Cette commande retourne entre autres, des indications similaires à la Figure 3.

À chaque attribut est associé un type, un mode de mise à jour, une valeur brute (RAW), une valeur WORST, une valeur THRESH et une valeur VALUE.

La valeur brute représente la valeur mesurée de l'attribut. VALUE représente un indice de fiabilité actuelle de cet attribut. THRESH représente la valeur limite avant une dégradation des performances et un risque de panne élevé : si $VALUE \leq THRESH$ alors le disque risque de tomber en panne. Les valeurs de THRESH sont définies par le constructeur du disque dur pour un modèle donné. WORST représente la plus petite valeur de VALUE enregistrée.

Le type d'attribut *Old-age* indique que si $VALUE < THRESH$ alors le produit est en fin de vie du fait d'une usure normale. L'attribut *Pre-Fail* indique que si $VALUE < THRESH$, alors une panne est imminente.

Le champ `WHEN_FAILED` indique la probabilité de panne : *FAILING_NOW* indique une panne imminente, *In_the_past* indique que l'indice VALUE est déjà passé une fois sous la valeur THRESH et donc

qu'il faudra surveiller ce disque de près. L'absence de valeur indique que le disque fonctionne normalement.

Pour effectuer un test rapide sur le disque afin d'y déceler des erreurs, vous pouvez lancer la commande suivante :

```
# smartctl -t short /dev/hda
```

L'option `short` peut être remplacée par `long`, qui génère un test plus approfondi. Et pour prendre connaissance des résultats du test, il faut lancer la commande :

```
# smartctl -t selftest /dev/hda
```

Cette commande permet de savoir rapidement si le test s'est effectué ou non sans erreur.

Aussi, la commande `smartctl -l error /dev/hda` permet de lister et d'identifier les erreurs détectées. Ici, veuillez à relativiser le résultat : quelques erreurs sur un disque en activité depuis plusieurs heures n'a rien d'alarmant.

Température et vitesse de ventilation

Les cartes mères modernes implémentent des composants électroniques permettant de connaître la température, la vitesse de rotation des ventilateurs ainsi que les tensions de l'alimentation. Le programme `lm_sensors` (à installer via votre gestionnaire de package) va permettre d'accéder à toutes ces informations.

La commande `sensors-detect` va nous permettre de lancer l'auto-détection des composants disponibles sur votre carte mère. Une série de questions va vous être posée. Répondez YES à chacune (ce qui ne veut pas dire qu'il ne faut pas lire ce qui vous est demandé !). En résumé, le système vous demande quel type de bus doit être scanné (PCI, I2c, SMBus) pour la recherche des détecteurs.

Ensuite, il vous faudra charger les modules adéquats, autrement dit, les modules que la commande `sensors-detect` vous aura indiqués, via la commande `modprobe`. Par exemple :

```
# modprobe i2c-sensor i2c-viapro i2c-isa it87
```

```
it87-isa-0290
Adapter: ISA adapter
VCore 1: +1.52 V (min = +4.08 V, max = +4.08 V) ALARM
VCore 2: +2.51 V (min = +4.08 V, max = +4.08 V) ALARM
+3.3V: +6.59 V (min = +8.16 V, max = +8.16 V) ALARM
+5V: +5.00 V (min = +6.85 V, max = +6.85 V) ALARM
+12V: +12.22 V (min = +16.32 V, max = +16.32 V) ALARM
+12V: -18.89 V (min = +3.93 V, max = +3.93 V) ALARM
+5V: +3.33 V (min = +4.03 V, max = +4.03 V) ALARM
Stdbv: +5.00 V (min = +6.85 V, max = +6.85 V) ALARM
Vbat: +3.34 V
fan1: 2812 RPM (min = 0 RPM, div = 8)
fan2: 0 RPM (min = 0 RPM, div = 8)
fan3: 0 RPM (min = 0 RPM, div = 2)
M/B Temp: +38°C (low = -1°C, high = -1°C) sensor = thermistor
CPU Temp: +32°C (low = -1°C, high = -1°C) sensor = thermistor
Temp3: +33°C (low = -1°C, high = -1°C) sensor = diode

eeprom-12c-0-51
Adapter: SMBus Via Pro adapter at 0400
Memory type: DDR SDRAM DIMM
Memory size (MB): 256
```

Fig. 4

Enfin, vous pourrez tester vos détecteurs à l'aide de la commande `sensors`, qui devrait vous retourner quelque chose similaire à la Figure 4.

M/B Temp et CPU Temp correspondent respectivement aux températures de la carte mère et du processeur. Les lignes commençant par `fan` nous renseignent quant à la vitesse de rotation du ou des ventilateurs (en Rotations Par Minute).

Si vous souhaitez éviter de passer par la ligne de commande pour afficher ces informations, sachez que le bureau Gnome propose une *applet* nommée `sensors-applet`. Les utilisateurs de l'environnement KDE pourront quant à eux utiliser le *frontend* nommé `kensors`.

Comment est partitionné mon disque ?

Il peut être grandement utile de savoir comment est partitionné son disque si l'on souhaite faire cohabiter plusieurs OS ou plusieurs distributions. Votre environnement de bureau vous propose sûrement un utilitaire graphique pour visualiser le partitionnement de votre disque dur (comme un « gestionnaire de disques » ou assimilé). Néanmoins, il est très facile (et rapide !) de visualiser le partitionnement via la ligne de commande.

Ainsi, la commande `fdisk -l` permet de lister la table de partition du périphérique désigné. Si vous ne spécifiez pas de périphérique, `fdisk` utilisera le premier disque dur détecté.

Attention, soyez très prudent en utilisant cette commande, car elle est permet également de repartitionner votre disque si vous utilisez l'option correspondante.

De quelle quantité de mémoire je dispose ?

Pour combler un éventuel manque de mémoire vive, les systèmes Linux utilisent une mémoire « virtuelle » pour simuler un bloc de mémoire supplémentaire. C'est ce que l'on appelle la SWAP. La mémoire totale disponible pour le système est la somme des deux. C'est le noyau qui gère le stockage en RAM ou en SWAP, selon le cas.

La commande `free` affiche les quantités totales de mémoire RAM et SWAP, libre et utilisée, ainsi que la mémoire partagée et les tampons utilisés par le noyau. L'option `-m` permet de présenter le résultat en Mo. Ainsi, pour afficher à l'écran la RAM et la SWAP disponible sur votre système, entrez la commande :

```
$ free -mt
```

L'option `-t` permet de retourner la somme des deux types de mémoire. Vous devriez obtenir quelque chose comme ceci :

	total	used	free	shared	buffers	cached
Mem:	250	240	10	0	4	84
-/+ buffers/cache:	151	99				
Swap:	486	123	363			
Total:	737	363	373			

Fig. 5