

# Gestion des périphériques

- Objet : Détail de la gestion du matériel
- Niveau requis :  
  [avisé](#)
- Commentaires : *Le fonctionnement du système avec le matériel embarqué.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :  
  à, [compléter](#)
  - Création par [smolski](#) 29/11/2018
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#) <sup>1)</sup>

## Nota :

Ce tuto est la copie d'un cours proposé par raleur sur le forum ici :

- <https://debian-facile.org/viewtopic.php?pid=284497#p284497>

Que les pas de **raleur** soient menés sur un tapis de pétales de rose à jamais... 😊

## Introduction

Ce qui suit concerne plus particulièrement les périphériques “plug&play” connectés à un bus de type PCI ou USB.

## Fonctionnement

Il y a deux niveaux de communication entre le noyau et un périphérique :

1. le niveau bas “bus”
2. le niveau haut “fonctionnel”

## Le bus

La communication au niveau bus est commune à tous les périphériques connectés par un même type de bus : PCI, USB, SATA... Elle ne sert pas à faire fonctionner le périphérique mais à le détecter, l'identifier, énumérer ses ressources (plages mémoire et E/S, interruptions...), détecter sa déconnexion... Cette communication est assurée par le pilote du bus.

## La fonction

La communication au niveau fonctionnel est spécifique à chaque modèle de périphérique, ou à chaque classe de périphérique s'il s'agit d'un périphérique standard (clavier, souris, stockage de masse, contrôleur AHCI...). Cette communication est assurée par le pilote spécifique du périphérique

ou de la classe de périphérique, qui s'appuie sur le pilote du bus. Elle a pour but d'utiliser le périphérique et de le faire fonctionner.

## Périphériques et bus

Quand un périphérique est détecté sur un bus, il est d'abord interrogé par le pilote du bus pour l'identifier.

Le périphérique renvoie diverses informations sur lui-même, dont plusieurs identifiants :

1. identifiant constructeur (propre à chaque type de bus)
2. identifiant produit (propre à chaque constructeur)
3. identifiant de classe (propre à chaque type de bus)

Ce sont par exemple les informations qu'on peut afficher avec [lspci](#) ou [lsusb](#).

On peut aussi les retrouver dans l'arborescence de `/sys/bus/`.

## Le noyau

Le noyau va ensuite notifier divers composants logiciels du système qui se sont enregistrés pour recevoir ces informations :

1. les pilotes compilés en dur dans le noyau
2. les pilotes en modules déjà chargés
3. le démon [udev](#)

## Pilote & Module

Lorsqu'un pilote du noyau compilé en dur ou en module chargé reconnaît les identifiants d'un périphérique présent qu'il sait gérer, il s'associe à lui et le prend en charge. D'autre part, chaque pilote plug&play compilé en module contient une liste de combinaisons d'identifiants de périphériques qu'il sait gérer sous forme d'alias. C'est ce qui est affiché par *modinfo* dans les lignes "alias:" pour un module donné.

Il peut s'agir par exemple de couples d'identifiants constructeur+produit pour le pilote d'un périphérique particulier, ou d'identifiants de classe pour le pilote d'un périphérique générique.

## udev

Lorsqu'udev est notifié par le noyau de la détection d'un périphérique, il exécute *modprobe* avec la combinaison d'identifiants renvoyés.

## Modprobe

*Modprobe* recherche une correspondance avec un module dans le fichier `/lib/modules/$(uname -r)/modules.alias`, qui a été créé par *depmod* en scannant l'ensemble des modules présents

dans `/lib/modules/$(uname -r)` lors de l'installation du noyau.  
Si une correspondance est trouvée, le module est chargé et prend le périphérique en charge.

1)

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

From:

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

Permanent link:

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



Last update: **29/11/2018 03:58**