

# Administration Systèmes & Réseaux

## Technologies de gestion de stockage

May 3, 2015

Houcemeddine HERMASSI

`houcemeddine.hermassi@enit.rnu.tn`

École Nationale d'Ingénieurs de Carthage ENI-CAR  
Université Carthage  
Tunisie



Représentation des disques

Les partitions d'un disque

SWAP

### IDE

Les disques reliés à des contrôleurs IDE (appelés aussi PATA, Parallel Ata, ou ATAPI) se nomment hdX :

- ▶ hda : IDE0, Master
- ▶ hdb : IDE0, Slave
- ▶ hdc : IDE1, Master
- ▶ hdd : IDE1, Slave
- ▶ etc

Contrairement aux idées reçues, il n'y a pas de limites au nombre de contrôleurs IDE, sauf le nombre de ports d'extension de la machine (slots PCI). De nombreuses cartes additionnelles existent, de nombreuses cartes mères proposent jusqu'à quatre, six, huit connecteurs. Dans ce cas, les fichiers se nomment hde, hdf, hdg, etc.

Les lecteurs CD-Rom, DVD et graveurs sont vus comme des disques IDE et respectent cette nomenclature.

Les derniers noyaux Linux utilisent par défaut une API appelée libata pour accéder à l'ensemble des disques IDE, SCSI, USB, Firewire, etc. Si c'est votre cas (regardez les notes de version de la distribution), la nomenclature reprend celle des disques SCSI, abordée au point suivant.

# Représentation des disques

## Nomenclature



### SCSI, SATA, USB, FIREWIRE, etc.

Les disques reliés à des contrôleurs SCSI, SCA, SAS, FiberChannel, USB, Firewire (et probablement d'autres interfaces exotiques comme les lecteurs ZIP sur port parallèle) se nomment sdX. L'énumération des disques reprend l'ordre de détection des cartes SCSI et des adaptateurs (hosts) associés, puis l'ajout et la suppression manuelle via hotplug des autres.

- ▶ sda : premier disque SCSI
- ▶ sdb : deuxième disque SCSI
- ▶ sdc : troisième disque SCSI
- ▶ etc

La norme SCSI fait une différence entre les divers supports. Aussi les lecteurs CD-Rom, DVD, HD-DVD, BlueRay et les graveurs associés n'ont pas le même nom. Les lecteurs et graveurs sont en srX (sr0, s r 1 , etc.). Vous pouvez aussi trouver scd0, scd1, etc. mais ce sont généralement des liens symboliques vers sr0, s r 1 , etc.

La commande **ls SCSI** permet d'énumérer les périphériques SCSI.

```
$ ls SCSI
[4:0:0:0]    disk      ATA          ST380011A      8.01  /dev/sda
[5:0:0:0]    cd/dvd    LITE-ON      COMBO SOHC-4836V S9C1  /dev/sr0
[31:0:0:0]   disk      USB2.0       Mobile Disk     1.00  /dev/sdb
```

# Les partitions d'un disque

## Découpage logique



### Principe

Un disque peut être vu comme une longue bande d'espace de stockage découpée en cases pouvant contenir une quantité donnée d'informations. Le disque peut être utilisé tel quel comme espace de stockage, rien n'empêche de créer un système de fichiers sur un disque sans passer par l'étape de partitionnement. Il est cependant important de donner une organisation logique à cet espace et aux systèmes de fichiers qu'il contiendra, ne serait-ce qu'au nom de la séparation des données (les fichiers de données) et des traitements (les programmes les utilisant et le système). La partitionnement consiste en un découpage logique du disque. Le disque physique, réel, est fractionné en plusieurs disques virtuels, logiques, les partitions. Chaque partition est vue comme un disque indépendant et contient son propre système de fichiers.

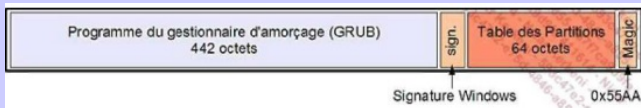
# Représentation des disques

## Organisation d'un disque



### MBR

Le premier secteur est le MBR, Master Boot Record, ou zone d'amorce. D'une taille de 512 octets il contient dans ses 444 premiers octets une routine (un programme) d'amorçage destiné soit à démarrer le système d'exploitation sur la partition active, soit un chargeur de démarrage (bootloader), puis 4 octets d'une signature optionnelle (Windows), 2 octets nuls, et les 64 octets suivants contiennent la table des quatre partitions primaires. Le tout finit par une signature 0xAA55 sur deux octets.



# Représentation des disques

## Organisation d'un disque



### Les partitions

Une partition est un découpage logique du disque. Il en existe trois sortes :

- ▶ Les partitions primaires, au nombre de quatre, sont celles décrites dans le MBR.
- ▶ Les partitions étendues (primaires étendues), une seule par disque (bien que théoriquement il soit possible de créer des partitions étendues au sein d'autres partitions étendues).
- ▶ Les partitions ou lecteurs logiques.

Un disque peut contenir jusqu'à 63 partitions en IDE, 15 en SCSI (c'est une limite de l'implémentation officielle du SCSI) ou via la libata. La limite actuelle est de 15 partitions pour tous les disques avec les derniers noyaux et l'API libata. Cependant quelques distributions permettent d'utiliser l'ancienne API (PATA) pour revenir à l'ancien système.

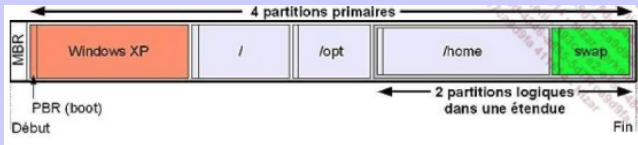
### Les partitions

Les partitions sont numérotées de 1 à n (15 ou 63). Une partition d'une valeur supérieure ou égale à 5 indique qu'il s'agit forcément d'une partition logique. Comme il ne peut y avoir que quatre partitions primaires, la dernière (la 4) est souvent créée comme étendue :

- ▶ Partitions 1 à 3 : primaires
- ▶ Partition 4 : étendue
- ▶ Partitions 5 à n : logiques

Le numéro de la partition apparaît à la suite du nom du fichier périphérique de disque :

- ▶ hda1 : première partition primaire du premier disque IDE
- ▶ hdb5 : cinquième partition, première partition logique du second disque IDE
- ▶ sda3 : troisième partition primaire du premier disque SCSI / libata
- ▶ sdc8 : huitième partition, soit quatrième partition logique du troisième disque SCSI/libata.





### EBR

Chaque partition étendue devant décrire les partitions logiques qu'elle contient, elle doit aussi disposer d'une table de partition. L'EBR (Extended Boot Record) reprend la structure du MBR sauf qu'il n'y a que deux enregistrements possibles dans la table des partitions. Le premier indique effectivement la position et la taille d'une partition logique, tandis que le second est vide si c'est la seule partition logique, ou pointe sur un autre EBR. Il peut donc y avoir plusieurs EBR dans une partition étendue.

- ▶ Les EBR forment une liste chaînée, la seconde entrée de partition pointant sur l'EBR suivant.
- ▶ Il n'y a qu'une seule partition logique décrite par EBR.

# Représentation des disques

## Organisation d'un disque



### PBR

Le PBR (Partition Boot Record), aussi appelé VBR (Volume Boot Record) ou Partition Boot Sector est le premier secteur de chaque partition primaire ou logique. Il peut contenir une routine de démarrage d'un système d'exploitation, un chargeur de démarrage, voire rien du tout si la partition n'a pas vocation à être bootée. Quand le MBR ne contient pas de routine, le bios tente de démarrer et d'exécuter la routine du PBR de la partition marquée active.

# Représentation des disques

## Manipuler les partitions

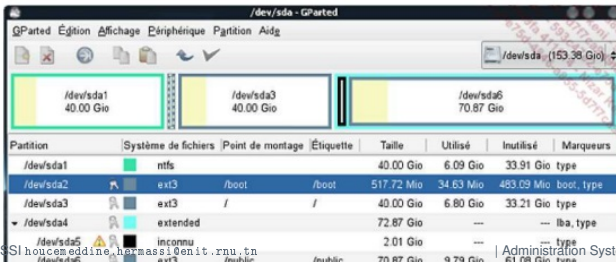


### Outils disponibles

Les outils **fdisk**, **cfdisk**, **sfdisk** ou encore **parted** permettent de manipuler les partitions

- ▶ **fdisk** est le plus ancien et le plus utilisé des outils de partitionnement. Il n'a aucun rapport avec le fdisk de Microsoft. Il est à base de menus et raccourcis textuels.
- ▶ **cfdisk** est un peu plus « visuel » et s'utilise avec les flèches directionnelles. Il permet les mêmes opérations que fdisk mais de manière plus conviviale.
- ▶ **sfdisk** fonctionne en interactif ou non, est assez compliqué mais plus précis.
- ▶ **parted** permet des opérations très avancées sur les partitions comme par exemple leur redimensionnement. Il est soit interactif (c'est un interpréteur de commandes) soit scriptable. Il existe des interfaces graphiques comme qtparted ou gparted.

Vous pouvez voir sur la capture gparted en action.



# Représentation des disques

## Manipuler les partitions



### Lister

C'est l'outil fdisk, à la fois le plus ancien et le plus standard, qui est généralement utilisé par les administrateurs et les ingénieurs système. Fdisk se lance en tant que root.

```
fdisk [-l] [disque] "
```

### Lister des partitions

L'exemple suivant se base sur un disque reconnu par le système comme /dev/sdb, et ne contenant aucune partition. Le but est de créer trois partitions : une primaire, une étendue et une logique.

1- Lancez fdisk avec le disque en argument, ne tenez pas compte des premières lignes affichées sauf si elles indiquent une erreur.

```
# fdisk /dev/sdb "
```

2- Vérifiez tout d'abord l'existence de partitions avec la touche p (print) puis [Entrée].

### Sauver

Quittez fdisk en sauvant votre table des partitions avec la touche w (write). Fdisk écrit la nouvelle table des partitions dans le MBR et/ou les EBR. Vous risquez d'obtenir des avertissements ici indiqué en gras.

```
Commande (m pour l'aide): w "
```

# Représentation des disques

## Manipuler les partitions



### Créer

Pour créer une partition, utilisez la touche n (new). Vous devez ensuite choisir le type de partition : primaire ou étendue.

Commande (m pour l'aide): n "

### Supprimer

Pour supprimer une partition, utilisez la touche d (delete) puis, si plusieurs partitions sont présentes, le numéro de partition (sdbX, X étant le numéro). Si une seule partition est présente, elle est prise par défaut.

Commande (m pour l'aide): d " *AVERTISSEMENT: la re-lecture de la table de partitions a échoué avec l'erreur 16: Périphérique ou ressource occupé. Le kernel va continuer d'utiliser l'ancienne table.*

La nouvelle table sera utilisé lors du prochain réamorçage. "

Synchronisation des disques. "

Ce message signifie que, comme le périphérique disque est en cours d'accès ou d'utilisation, Linux ne voit pas encore la nouvelle table et donc les nouvelles partitions créées. Ceci peut être confirmé avec la commande suivante. Notez que la dernière ligne devrait vous montrer votre nouvelle partition, ce qui n'est pas le cas.

```
# cat /proc/partitions | grep sdb "  
8 16 3932160 sdb "
```

# Représentation des disques

## Manipuler les partitions



### Forcer la synchronisation

Pour corriger ce dernier problème et forcer le noyau à relire la table des partitions, vous avez le choix entre deux commandes. La première est **blockdev** avec le paramètre `-rereadpt` (re-read partition table).

```
# blockdev -rereadpt /dev/sdb "
```

La seconde est **partprobe**, disponible seulement si parted est installé. Elle peut réussir si blockdev a échoué. Par défaut elle relit les tables de toutes les partitions, mais vous pouvez lui spécifier le disque en argument.

```
# partprobe /dev/sdb "
```

Vérifiez si la partition est bien reconnue.

```
# cat /proc/partitions | grep sdb "
```

### Pourquoi créer un swap?

Dans un environnement 32 bits un processus peut théoriquement accéder à 4 Go d'espace mémoire. Il dispose de 4 Go de mémoire virtuelle, rien qu'à lui et à laquelle aucun autre processeur ne peut accéder. Dans la pratique il y a plusieurs freins à cette possibilité :

- ▶ L'espace mémoire adressable d'un processus est partagé entre zone de code et zone de données dont la taille peut varier selon le noyau utilisé.
- ▶ Les ordinateurs ne disposent pas tous de 4 Go de mémoire (bien qu'il soit courant de trouver des serveurs Linux disposant de 16, 32 ou même 64 Go de mémoire).
- ▶ Tous les processus doivent se partager la mémoire de l'ordinateur.

Que se passe-t-il si un processus n'a plus assez de mémoire pour traiter ses données ? Le système d'exploitation va décharger des segments de la mémoire physique dans une zone d'échange sur disque qui fera office de mémoire virtuelle tampon. Il y a donc un échange entre la mémoire physique et cette zone d'échange, appelé l'espace de swap. Ce processus permet d'utiliser plus de mémoire que l'ordinateur n'en dispose réellement, au prix d'un net ralentissement si le programme est très gourmand.

### Taille optimale

Il n'y a pas de règles strictes sur la taille du swap. Cependant les quelques règles courantes suivantes sont valables dans la plupart des cas :

- ▶ Moins de 512 Mo de RAM : deux fois la RAM.
- ▶ 1 Go à 4 Go : la taille de la RAM.
- ▶ Plus de 4 Go : 4 Go, plus ou moins, selon l'utilisation des processus.

### Créer une partition swap

1- Vous savez déjà créer une partition. Créez une partition avec fdisk de la taille souhaitée pour le swap, et donnez- lui le type 83 .

2- Synchronisez la table des partitions avec **partprobe**.

3- Utilisez la commande **mkswap** pour préparer la partition à recevoir du swap.

```
# mkswap /dev/sda5 "
```

```
Initialisation de la version de l'espace de swap 1, taille = 2154983 kB "  
pas d'étiquette, UUID=c84714e6-c42c-44d4-9fe7-10dc6afac644"
```

4- Votre swap est prêt.

Il est possible d'attribuer une étiquette à la partition de swap avec le paramètre -L.



### Activation dynamique

Linux permet d'activer et de désactiver le swap, ou des morceaux de swap, directement sans avoir à redémarrer le système.

La commande **swapon** permet d'activer une partition de swap :

```
# swapon /dev/sda5 "
```

Le paramètre -p permet de modifier la priorité du swap. Plus la valeur, comprise entre 0 et 32767, est élevée, plus la priorité d'une zone de swap est élevée. Le système l'utilisera en priorité. Ce paramètre est utile si plusieurs partitions de swap existent sur des disques différents. Dans ce cas, privilégiez soit le disque le plus rapide, soit indiquez une priorité égale pour une meilleure répartition.

Comme avec mount, le paramètre -L permet d'activer une zone de swap grâce à son étiquette.

La commande **swapon** désactive une zone de **swap**. Veillez à disposer de l'espace mémoire libre nécessaire, si non la commande ne fonctionnera pas.

Le contenu de **/proc/swaps** reflète l'état actuel des zones de swap actives.

```
# cat /proc/swaps
```

Filename	Type	Size	Used	Priority
/dev/sda5	partition	1461872	2012	-1

### Dans /etc/fstab

Les zones de swap se placent dans le fichier /etc/fstab. Voici un exemple :

```
/dev/sda5 swap swap defaults 0 0 "
```

Les options **noauto** et **pri=X** peuvent être précisées. L'option **pri** permet de définir la priorité de la zone de swap. Lors du démarrage, le système exécute **swapon -a** qui active toutes les partitions de swap présentes dans la fsta sauf si noauto est précisé.

Lors de l'arrêt, le système exécute **swapoff -a** qui désactive complètement le swap.

```
# cat /proc/swaps
```

Filename	Type	Size	Used	Priority
/dev/sda5	partition	1461872	2012	-1

## Fichier de swap

Voici les manipulations pour un petit swap de 32 Mo:

```
# free | grep Swap
Swap:      2104472      4344      2100128
# dd if=/dev/zero of=/swap bs=1024 count=32768
32768+0 enregistrements lus
32768+0 enregistrements écrits
33554432 bytes (34 MB) copied, 0,35697 s, 94,0 MB/s
slyserver:~ # mkswap /swap
Initialisation de la version de l'espace de swap 1, taille = 33550 kB
pas d'étiquette, UUID=b2e5e99e-09a1-4b2d-ac76-59f76526453a
slyserver:~ # chmod 600 /swap
slyserver:~ # sync
slyserver:~ # swapon -v /swap
swapon sur /swap
slyserver:~ # free | grep Swap
Swap:      2137232      4308      2132924
```

Modifiez éventuellement le fichier `/etc/fstab`, en espérant que le swap soit activé après le montage des systèmes de fichiers. Le swap est activé au boot, généralement après le montage de `/`.

```
/swap swap swap defaults 0 0 "
```

### Free

La commande **free** vous fournit des informations sur la mémoire physique (RAM) de votre ordinateur ainsi que sur l'occupation du swap. L'unité par défaut est le Ko (identique avec le paramètre -k), mais elle peut être modifiée en Mo (-m) voire en Go (-g).

```
# free -k
```

	total	used	free	shared	buffers	cached
Mem:	2062484	2045732	16752	0	707512	776528
-/+ buffers/cache:		561692	1500792			
Swap:	2104472	132	2104340			

### Mémoire réservée

2 Go correspondent à 2097152 Ko. Or le total présente une différence d'environ 34 Mo. Cette mémoire est réservée au noyau et ne peut pas être utilisée par d'autres programmes. Elle est utile pour les traitements du noyau, son chargement, l'initrd. Voyez le résultat de la commande suivante (volontairement tronqué) :

```
# dmesg | grep -i memory
...Memory: 2057756k/2096640k available (2053k kernel code, 38496k
reserved, 1017k data, 316k init)
Freeing initrd memory: 441k freed
Freeing unused kernel memory: 316k freed
...
```

Le système se réserve environ 38 Mo de mémoire, puis libère la mémoire dont il n'a plus besoin, pour obtenir le résultat attendu.

### meminfo

Le système de fichiers virtuel `/proc` contient des informations détaillées sur la mémoire au travers du pseudo- fichier `/proc/meminfo`. Il semble bien difficile de trouver quelque chose de plus complet. La sortie suivante est un résultat sur un système Linux en 64 bits. En 32 bits, deux lignes supplémentaires (`highmem` et `lowmem`) indiquent les zones réservées aux données et au noyau. Les premières lignes et les lignes concernant le swap sont identiques au résultat de la commande `free`.

```
# cat /proc/meminfo "
```

Merci pour votre attention!

