

# Introduction à GNU/Linux

## Disques

Pascal Bessonneau

Starinix

11/2017

## Les dossiers personnels

### Montage et démontage

- Ajouter le montage d'un disque au démarrage

- Gestion des disques en mode graphique

- Gestion des disques en mode graphique

- Ajouter le montage d'un disque au démarrage

### Les autres répertoires

- Les devices

- Les autres répertoires

### Les formats de partition

- gparted

### Les permissions

- permission

### Utilitaires

## Le home directory *GNU/Linux*

Les arborescences de fichiers *GNU/Linux* sont souvent source de confusion.

Pourtant ils ne sont on ne peut plus simple. . .

Votre *home directory* ou répertoire personnel est dans le répertoire */home/utilisateur* avec utilisateur votre login.

Vous y passerez la quasi totalité de votre temps.

Ce répertoire s'appelle aussi `~` pour le répertoire de l'utilisateur actif. On peut écrire pour les utilisateurs `~utilisateur`.

# Le home directory *GNU/Linux*

Tous les utilisateurs sont dans la même situation sauf le super utilisateur (ou root).

## Les périphériques amovibles

Quand vous montez une clef usb par contre (ou un CD), le système ne va pas la monter dans votre répertoire personnel. Il va la monter dans le répertoire */media/utilisateur/nom de la clef*. Les distributions plus vieilles montent dans le répertoire */mnt/*. Par monter, on entend rendre possible l'accès du périphérique à l'utilisateur.

# Montage et démontage

On a parlé de montage et démontage :

1. quand on branche un périphérique, il est reconnu par le système d'exploitation mais pas accessible par l'utilisateur
2. il faut lui affecter un répertoire (et son type de format) pour qu'il soit accessible
3. cette opération se fait par la commande *mount*. Elle nécessite parfois d'être le super-utilisateur

# Montage et démontage

Par exemple pour monter une clef « à la main » :

```
mount -t vfat /dev/sdg /home/pascal/clef
```

Les deux éléments les plus importants sont :

1. /dev/sdg : c'est la référence vers le matériel ici une partition
2. /home/pascal/clef : c'est le répertoire dans lequel le contenu de la clef est visible. Attention si il y a des choses dans ce répertoire, ces éléments sont "masqués" tant que la clef est monté.

# Avoir l'UUID

Il faut pour ça utiliser avec sudo l'utilitaire *blkid* :

```
/dev/sda1: UUID="8bf33340-e94c-..." TYPE="ext4"
```

```
/dev/sda2: UUID="ac56a704-260b-..." TYPE="swap"
```

```
/dev/sda3: LABEL="Home" UUID="8244710a-5cce-49ad-8b93-a92b5"
```

```
/dev/sda4: UUID="DCF041AFF0419126" TYPE="ntfs"
```



## Avoir l'UUID

Cette commande permet donc d'avoir l'UUID et d'identifier les disques et le type de format.

L'UUID est un identifiant unique qui, sauf formatage, restera identique « à vie ».

Si on utilise le nom du périphérique, par exemple `/dev/sda1`, pour identifier un disque plutôt que le numéro unique, en cas de changement de configuration, le nom de périphérique risque de changer contrairement à l'UUID.

# Sous GNOME 3

Il y a un utilitaire *Disques*.

# Sous GNOME 3

Il y a un utilitaire *Disques*.

# Sous GNOME 3

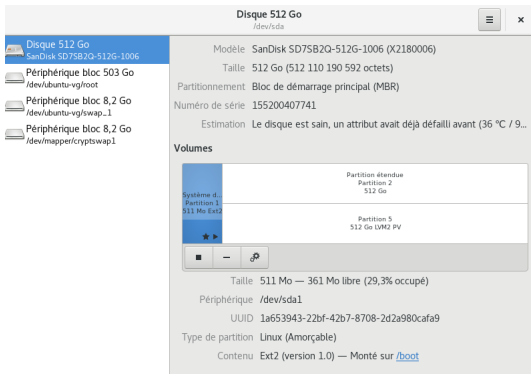


FIGURE: Disques sous GNOME

# Sous GNOME 3

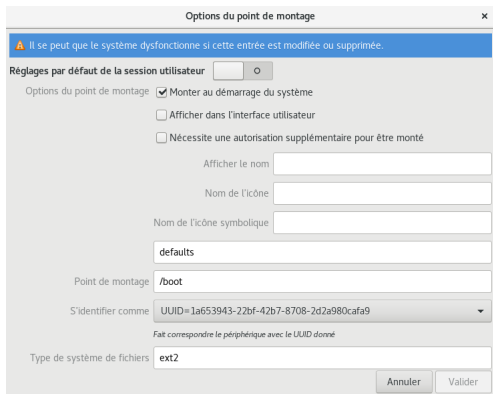


FIGURE: Disques sous GNOME

## /etc/fstab

Comme on a vu précédemment, on utilise l'UUID dans ce fichier de configuration plutôt que le nom de périphérique.

```
UUID=c3cc32c0-b4bd-... /boot ext2 defaults 0 2
```

## /etc/fstab

Attention, au démarrage, le système risque de se bloquer si le disque est marqué comme à monter par défaut et qu'il n'est pas présent. Avant d'enlever un disque penser à commenter la ligne (ou la supprimer) avant le redémarrage.

```
UUID=c3cc32c0-b4bd-... /boot ext2 defaults 0 2
```

## /etc/fstab

Les arguments sont :

1. l'identifiant ou le nom du périphérique
2. le point de montage
3. le type de la partition
4. des mots clefs pour changer les propriétés du disque
5. le *dump*, utilisé pour les sauvegardes
6. le *pass*, pour les vérifications on démarrage



## /etc/fstab

Parmi les mots-clefs à connaître :

**user, no user** définit si un utilisateur (et pas seulement le root) a le droit de monter la partition

**auto,noauto** définit si la partition est montée automatiquement au démarrage et en faisant *mount -a*

**atime/noatime** définit si le système marque la date de dernier accès. Il faut le mettre de préférence à *noatime* pour les SSD

**rw/ro** montage en lecture/écriture ou lecture seule (read only)

**uid,gid,...** permet de spécifier les droits des fichiers du disque

## /etc/fstab

Pour le *dump*, il faut le laisser à 0.

Pour le *pass*, les valeurs sont à indiquer sont les suivantes :

1. pour la racine
2. pour les autres partitions Linux
3. pour le swap et les partitions windows

Plus d'infos là

# /etc/mtab

## Le répertoire `/dev/`

Il y a un dicton qui dit que tout dans *GNU/Linux* est fichier. Ce qui traduit l'existence du répertoire `/dev/`. A une entrée dans ce répertoire correspond un matériel ou une fonctionnalité du noyau. Par exemple avec la commande `sudo blkid` vous pouvez voir les répertoires `/dev/` correspondant aux disques. Un autre exemple est `/dev/urandom` qui contient des données aléatoires générées par le noyau.

```
dd bs=4M count=1 if=/dev/urandom of=random.txt
```

## Le répertoire /dev/

Les fichiers respectent des nomenclatures. Par exemple, *sdX* désigne un disque SCSI (historiquement).

*/dev/sdg* désigne le disque tout entier, */dev/sdg1* la partition 1 du disque *sdg*.

Un utilitaire en ligne de commande pour explorer les disques est *fdisk* ou plus récent *parted*.

Graphiquement utiliser *gparted*.

## Les principaux répertoire. . .

- /bin, la plupart des programmes en ligne de commande
- /dev, pointe vers les périphériques
- /etc, Contient les fichiers de configuration du système
- /lib, les librairies
- /tmp, pour les répertoires temporaires
- /usr, là où s'installe la plupart des programmes
- /var, contient les informations partagée (par exemple site web, logs, . . .)

La liste complète est là.

## /bin

Dans ce répertoire on trouve les binaires c'est-à-dire les programmes exécutables.

Vous pouvez également trouver des exécutables dans les répertoires */usr/share/bin* et */usr/local/bin*.

La disposition des exécutables dépend de la distribution et du type d'exécutables.

## /etc

Si vous voulez modifier la configuration système, il est probable que vous ayez à intervenir dans ce répertoire.

Les fichiers sont presque tous des fichiers texte qu'il suffit d'éditer en tant que super utilisateur.



# /lib

C'est le répertoire dans lequel on trouve les librairies c'est-à-dire des « morceaux » de programme qui sont mis dans le pot commun pour plusieurs programmes.

## /tmp

Dans ce répertoire on trouve tous les fichiers qui sont destinés à avoir une durée de vie courte. D'ailleurs sur la plupart des distributions, le répertoire est vidé quand le système démarre.

## /usr

Dans ce répertoire on trouve pas mal de choses. Il y a aussi des exécutables des librairies, l'emplacement de tout ça est régi par des règles dépendant de la distributions et de l'histoire de *GNU/Linux*. Dans ce répertoire on trouve surtout les exécutables qui sont lancés en mode graphique.

## /var

Ce répertoire est important notamment car vous trouverez par exemple deux répertoires très précieux :

- /var/www, le contenu de votre site web
- /var/logs, les fichiers journaux qui stockent les évènements qui se produisent sur le poste.

# /var

La page est assez exhaustive.  
Elle contient pas d'informations intéressantes.

# Les formats extX

Particularité des systèmes de fichier  
sortie de fdisk -l

# Les formats extX

```
file -sL /dev/sd*
```

# Les formats extX

**ext2** c'est le format standard UNIX qui n'est plus au goût du jour

**ext3** il reprend le format ext2 avec une journalisation

**ext4** il reprend ext3 avec en plus une augmentation de la taille des disques

A l'heure actuelle, il faut préférer le format le plus récent ext4.



# Les formats Windows

Si vous voulez monter des disques Windows :

**FAT,FAT32** c'est le vieux format qu'on retrouve sur les clefs USB notamment. Il est compatible *Linux*, Mac OS et Windows

**NTFS** c'est le « nouveau » format de Windows qui est propriétaire

Avant *Linux* ne pouvait écrire que sur des partitions FAT mais maintenant il peut aussi écrire sur des partitions NTFS dans la plupart des distros (mais pas toutes!).

# Les formats Windows

Il y a d'autres types de partitions proposées : reiserfs, xfs, zfs, ...  
Il ne se sont pas installés et/ou sont propriétaires donc ne sont pas  
lus par toutes les distributions.

# gparted

## les sorties d'un ls

```
-rw-r--r-- 1 root root 1426 nov. 26 2016 debug
-rw-r--r-- 1 root root 1735 nov. 26 2016 dhclient.conf
drwxr-xr-x 2 root root 4096 nov. 1 12:29 dhclient-enter-h
drwxr-xr-x 2 root root 4096 nov. 1 12:29 dhclient-exit-ho
```

## les sorties d'un ls

Les premiers caractères sont les droits pour le propriétaire et le groupe attribué au fichier ou au répertoire :

**d** directory ou répertoire.

**r** on peut lire le fichier

**w** on peut écrire le fichier

**x** le fichier est exécutable ou le répertoire peut être traversé

si au lieu d'un de ces caractères est remplacé par un - alors la propriété est inversé : si au lieu d'un w on a -, alors on ne peut pas écrire dans le fichier.

# Les différents utilisateurs

On peut voir que le trio rwx est répété trois fois.

Car il y a trois types d'utilisateurs :

1. le propriétaire du fichier ou *u(ser)*
2. un membre du groupe auquel est attaché le fichier ou *g(roup)*
3. tous les autres utilisateurs ou *empho(thers)*

## Dans l'explorateur de fichier

Avec un clic droit sur le fichier on peut changer avec nautilus ou dolphin, changer les droits : il suffit de choisir *Propriétés* puis *Permissions*.

En ligne de commande vous avez `chmod`. Il y a plusieurs utilisations de `chmod`.

# Utilisation simple de chmod

par exemple pour changer les propriétés d'un fichier, il faut faire :

```
chmod ug+rw fichier
```

cette commande permet de rendre lisible et en écriture *fichier*.

L'inconvénient est que la syntaxe ne permet de ne changer que par différence les permissions.



# Masque

Au lieu de modifier les permissions de façon relative, elle définit les droits mais sont moins intuitifs. A chaque catégorie rwx, est assigné un numéro et la somme définit les droits du fichiers.

- 1 exécution
- 2 écriture autorisée
- 4 lecture autorisée

En combinant les sommes de chacun de ces chiffres on obtient tous les combinaisons de droits.

# Masque

Ex : *chmod 754 fichier*

**7** =  $4 + 3 + 1$ , tous les droits sont donnés

**5** =  $4 + 1$ , lecture et exécution

**4** =  $4$ , lecture autorisée

et ainsi de suite.

# Récuratif

Pour changer les droits de tout un répertoire et ses sous répertoires, il suffit de faire *chmod 700 -R repertoire*

Attention, changer les droits des fichiers doit être fait de façon très prudente car il est difficile après une erreur de retrouver les droits de tous les fichiers d'une arborescence.

Et parfois le système peut être bloqué pour des problèmes de droit. Par exemple, vous bloquerez la mise à jour des fichiers de log si vous en changez les droits.

## Utilisateur et groupe

il y a deux commandes :

`chown pascal fichier` change le propriétaire de fichier pour pascal

`chown pascal :pascal fichier` change le propriétaire de fichier pour pascal et le groupe pour pascal

`chgrp pascal fichier` change le propriétaire de fichier pour pascal et le groupe pour pascal

# Utilitaires

Il y a quelques utilitaires qui n'ont pas d'équivalent en mode graphique.

Le premier est *df* qui permet d'avoir la place libre sur les disques. Il faut le lancer avec l'argument *df -h* pour avoir l'espace en unités intelligibles.

Il y a aussi la commande *du -ch* qui permet d'avoir l'espace disque pris par des répertoires.

Sous GNOME, un utilitaire peu efficace qui tente de les remplacer : *Analyseur d'utilisation de disques*