

# Linux : l'organisation de dossiers et la ligne de commande

Sergiu IVANOV

`sergiu.ivanov@u-pec.fr`

Les diapos disponibles en ligne :

`http://lacl.fr/~sivanov/doku.php?id=fr:  
cours\_de\_systemes\_et\_reseaux`

# Le prof est curieux

Qu'est-ce qu'un système d'exploitation ?

Qu'est-ce qu'un système de fichiers ?

Qu'est-ce que Linux ?

# Plan du cours

1. Les dossiers sous Linux

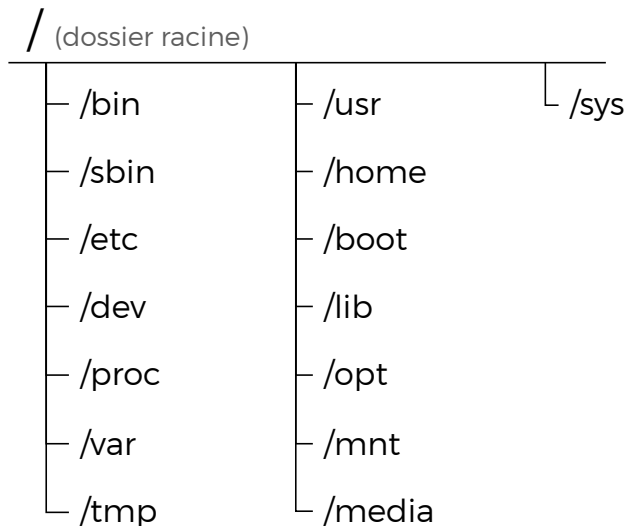
2. Introduction à la ligne de commande

# Plan du cours

## 1. Les dossiers sous Linux

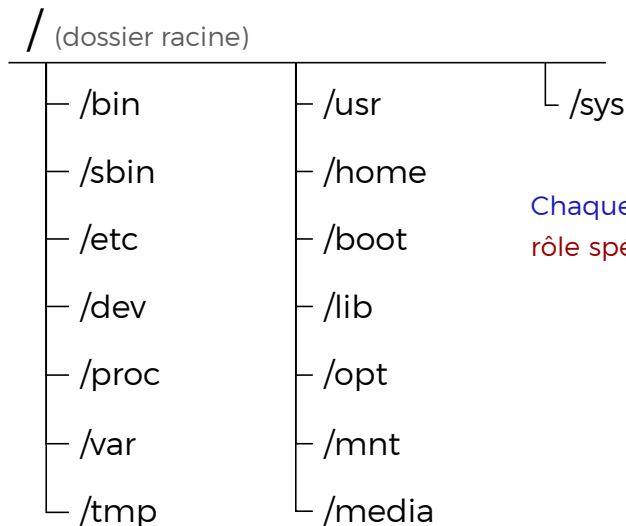
## 2. Introduction à la ligne de commande

# Structure typique de dossiers sous Linux



<http://www.thegeekstuff.com/2010/09/linux-file-system-structure/>

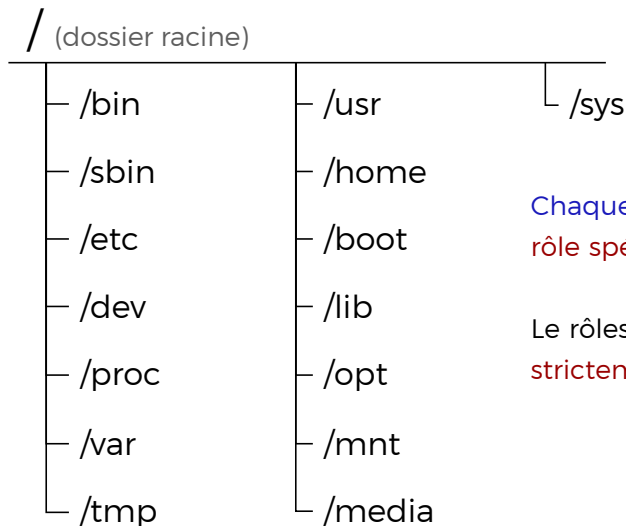
# Structure typique de dossiers sous Linux



Chaque dossier a un rôle spécifique.

<http://www.thegeekstuff.com/2010/09/linux-file-system-structure/>

# Structure typique de dossiers sous Linux

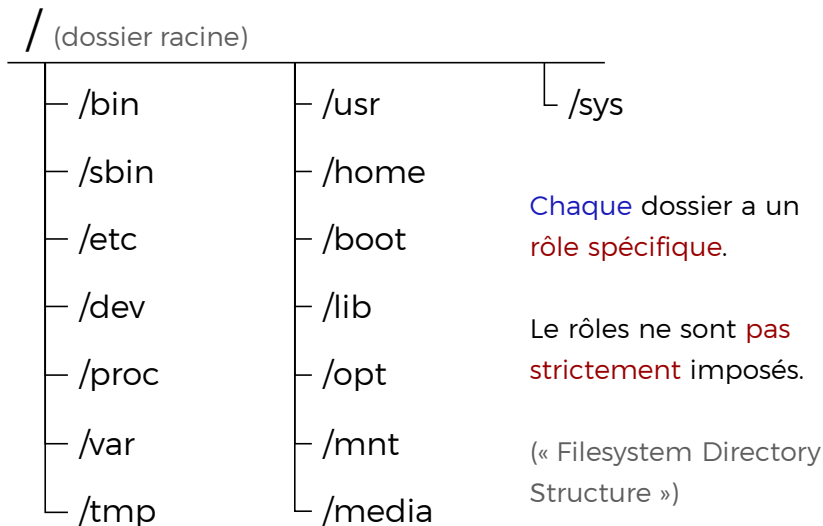


Chaque dossier a un rôle spécifique.

Le rôles ne sont pas strictement imposés.

<http://www.thegeekstuff.com/2010/09/linux-file-system-structure/>

# Structure typique de dossiers sous Linux



<http://www.thegeekstuff.com/2010/09/linux-file-system-structure/>



## Pause réflexion

**Pourquoi** avoir une telle **structure** ?  
(surtout si elle n'est pas strictement respectée)

## Pause réflexion

**Pourquoi** avoir une telle **structure** ?  
(surtout si elle n'est pas strictement respectée)

Avoir une **structure** de dossiers **détaillée** permet de ranger chaque chose **à sa place**.

## Pause réflexion

**Pourquoi** avoir une telle **structure** ?  
(surtout si elle n'est pas strictement respectée)

Avoir une **structure** de dossiers **détaillée** permet de ranger chaque chose **à sa place**.

La structure « **Filesystem Directory Structure** » est bien respectée en général, **sauf** dans des **cas exceptionnels**.

## Pause réflexion

**Pourquoi** avoir une telle **structure** ?

(surtout si elle n'est pas strictement respectée)

Avoir une **structure** de dossiers **détaillée** permet de ranger chaque chose **à sa place**.

La structure « **Filesystem Directory Structure** » est bien respectée en général, **sauf** dans des **cas exceptionnels**.

(plus de détails plus tard)

# / : dossier racine

- ▶ contient tous les autres dossiers
- ▶ n'est contenu dans aucun autre dossier
- ▶ un vrai dossier
  - ▶ on peut faire `cd /`

`/bin` : binaires utilisateur

binaire = fichier binaire exécutable

# /bin : binaires utilisateur

binaire = fichier binaire exécutable

- ▶ contient les binaires partagés par tous les utilisateurs du système
- ▶ contient donc toutes les commandes de la ligne de commande (presque)
- ▶ `cp`, `mv`, `ls`, `grep`

# /sbin : binaires système

- ▶ contient les **binaires** utilisés par l'**administrateur**
- ▶ `reboot`, `fdisk`, `iptables`



## /sbin : binaires système

- ▶ contient les binaires utilisés par l'administrateur
- ▶ reboot, fdisk, iptables

Qu'est-ce qu'un administrateur ?

# /sbin : binaires système

- ▶ contient les **binaires** utilisés par l'**administrateur**
- ▶ `reboot, fdisk, iptables`

## Qu'est-ce qu'un **administrateur** ?

L'**administrateur** est l'utilisateur qui a plus de **privilèges** pour pouvoir **administrer** le système :

- ▶ **installer/désinstaller** des programmes
- ▶ **changer les configurations**

# /etc : fichiers de configuration

- ▶ contient les fichiers de configuration générale
  - ▶ système
  - ▶ programmes

# /etc : fichiers de configuration

- ▶ contient les fichiers de configuration générale
  - ▶ système
  - ▶ programmes
- ▶ /etc/passwd
  - ▶ la liste d'utilisateurs avec leurs données
  - ▶ utilisé par le système

# /etc : fichiers de configuration

- ▶ contient les fichiers de **configuration** générale
  - ▶ **système**
  - ▶ **programmes**
- ▶ **/etc/passwd**
  - ▶ la **liste d'utilisateurs** avec leurs données
  - ▶ utilisé par le **système**
- ▶ **/etc/resolv.conf**
  - ▶ une **liste d'adresses** de serveurs qui transforment les **adresses symboliques** en **adresses IP**
  - ▶ utilisé par le **programme** qui récupère les adresses IP à partir des **adresses symboliques**

## /dev : fichiers de dispositifs

fichier de **dispositif** = fichier qui représente un **dispositif**

# /dev : fichiers de dispositifs

fichier de **dispositif** = fichier qui représente un **dispositif**

- ▶ **écrire** dans un fichier de dispositif = **envoyer** des données au dispositif
- ▶ **lire** d'un fichier de dispositif = **recevoir** des données du dispositif

# /dev : fichiers de dispositifs

fichier de **dispositif** = fichier qui représente un **dispositif**

- ▶ **écrire** dans un fichier de dispositif = **envoyer** des données au dispositif
- ▶ **lire** d'un fichier de dispositif = **recevoir** des données du dispositif

Exemples de fichiers de **dispositifs** (contenus dans **/dev**):

- ▶ **/dev/tty0** (teletype)
  - ▶ représente un **terminal** (~ une interface ligne de commande)
- ▶ **/dev/sda** (SCSI device a)
  - ▶ représente un **disque dur**



# /proc : information processus

- ▶ contient des informations sur
  - ▶ les ressources du système
  - ▶ les processus

# /proc : information processus

- ▶ contient des informations sur
  - ▶ les ressources du système
  - ▶ les processus
- ▶ `/proc/meminfo`
  - ▶ des information sur la mémoire vive :

```
MemTotal:          7873164 kB
MemFree:           506464 kB
...
```

- ▶ `/proc/[numéro]/`
  - ▶ des informations sur le processus avec l'identifiant `[numéro]`

# /proc : information processus

- ▶ contient des informations sur
  - ▶ les ressources du système
  - ▶ les processus
- ▶ /proc/meminfo
  - ▶ des information sur la mémoire vive :

```
MemTotal:      7873164 kB
MemFree:       506464 kB
...
```

- ▶ /proc/[numéro]/
  - ▶ des informations sur le processus avec l'identifiant [numéro]
- ▶ est un dossier virtuel
  - ▶ il n'existe pas sur le disque dur et est recréé à chaque démarrage du système

# /var : fichiers variables

- ▶ contient des fichiers qui **changent souvent**

# /var : fichiers variables

- ▶ contient des fichiers qui **changent souvent**
  - ▶ journaux (/var/log/)
  - ▶ bases de données (/var/db/)
  - ▶ courriels (/var/mail/)
    - ▶ sur les serveurs qui gèrent les courriels
  - ▶ tampons d'imprimantes (/var/spool/)
    - ▶ les données qui seront envoyées sur l'imprimante
  - ▶ verrous (/var/lock/)
    - ▶ les fichiers qui servent à gérer l'accès concurrent à des ressources partagées
  - ▶ fichiers temporaires (/var/tmp/)

## Astuce : le / à la fin de chemin

```
/var/tmp/
```

## Astuce : le / à la fin de chemin

`/var/tmp/`

- ▶ le chemin pointe vers un dossier
- ▶ notation très utilisée

## Astuce : le / à la fin de chemin

`/var/tmp/`

- ▶ le chemin pointe vers un dossier
- ▶ notation très utilisée
- ▶ ne pas confondre avec le dossier racine /



# /tmp : fichiers temporaires

- ▶ contient des fichiers temporaires (héhé)

# /tmp : fichiers temporaires

- ▶ contient des fichiers temporaires (héhé)
- ▶ est effacé à chaque redémarrage

# /tmp : fichiers temporaires

- ▶ contient des fichiers **temporaires** (héhé)
- ▶ est **effacé** à chaque **redémarrage**

**Note** : `/var/tmp/` contient aussi des fichiers **temporaires**, mais ce dossier **n'est pas effacé** au redémarrage.

# /lib : librairies système

- ▶ contient les **librairies** nécessaires pour les programmes dans `/bin/` et `/sbin/`

# /lib : librairies système

- ▶ contient les **librairies** nécessaires pour les programmes dans `/bin/` et `/sbin/`
- ▶ peut contenir des **modules noyau**

# /lib : librairies système

- ▶ contient les **librairies** nécessaires pour les programmes dans `/bin/` et `/sbin/`
- ▶ peut contenir des **modules noyau**
- ▶ les noms de fichiers ont généralement la forme `lib[nomlibrairie].so.[version]`
  - ▶ `libguile.so.17.4.0`

# /usr : programmes utilisateur

- ▶ contient les composants de programmes utilisateur

# /usr : programmes utilisateur

- ▶ contient les composants de programmes utilisateur
  - ▶ binaires (/usr/bin/)
    - ▶ less, man, date
  - ▶ binaires pour l'administrateur (/usr/sbin/)
  - ▶ librairies (/usr/lib/)
  - ▶ programmes compilés localement (/usr/local/)



# /usr : programmes utilisateur

- ▶ contient les composants de programmes utilisateur
  - ▶ binaires (/usr/bin/)
    - ▶ less, man, date
  - ▶ binaires pour l'administrateur (/usr/sbin/)
  - ▶ librairies (/usr/lib/)
  - ▶ programmes compilés localement (/usr/local/)

Les différences entre /bin/ et /usr/bin/, /sbin/ et /usr/sbin/, /lib/ et /usr/lib/ sont difficiles à tracer.

# /usr : programmes utilisateur

- ▶ contient les **composants** de programmes **utilisateur**
  - ▶ binaires (/usr/bin/)
    - ▶ less, man, date
  - ▶ binaires pour l'administrateur (/usr/sbin/)
  - ▶ librairies (/usr/lib/)
  - ▶ programmes compilés localement (/usr/local/)

Les différences entre /bin/ et /usr/bin/, /sbin/ et /usr/sbin/, /lib/ et /usr/lib/ sont **difficiles** à tracer.

La politique de gestion **varie** d'une distribution à l'autre.

# /opt : programmes optionnelles

- ▶ est là pour des raisons historiques
  - ▶ y avait quelqu'un qui l'utilisait sur un vieux système

# /opt : programmes optionnelles

- ▶ est là pour des raisons historiques
  - ▶ y avait quelqu'un qui l'utilisait sur un vieux système
- ▶ est généralement utilisé de nos jours pour éviter d'installer les binaires, les libraires, la documentation et le code source dans des dossiers différents
  - ▶ donc sert à contourner la « Filesystem Directory Structure »

# /opt : programmes optionnelles

- ▶ est là pour des raisons historiques
  - ▶ y avait quelqu'un qui l'utilisait sur un vieux système
- ▶ est généralement utilisé de nos jours pour éviter d'installer les binaires, les libraires, la documentation et le code source dans des dossiers différents
  - ▶ donc sert à contourner la « Filesystem Directory Structure »
- ▶ les applications propriétaires sont typiquement installées dans /opt/

# /boot : fichiers boot

boot = processus de démarrage  
de système

# /boot : fichiers boot

boot = processus de démarrage  
de système

- ▶ contient les fichiers nécessaires pour le démarrage
  - ▶ le noyau et quelques fichiers accessoires

# /sys : information système

- ▶ contient des informations sur le matériel et le système d'exploitation



# /sys : information système

- ▶ contient des informations sur le matériel et le système d'exploitation
- ▶ /sys/devices/
  - ▶ des informations sur les dispositifs connectés
- ▶ /sys/kernel/
  - ▶ des informations sur le noyau

# /home : dossiers personnels

- ▶ contient les **dossiers personnels** des utilisateurs
- ▶ `/home/tintin/`, `/home/tonton/`
- ▶ le dossier le plus simple à expliquer

# Supports de stockage accessoires

Où sont les **disques** ?

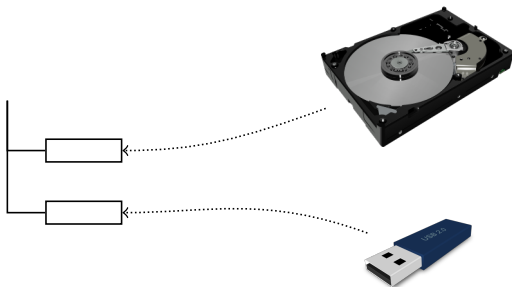
(question pour les utilisateurs Windows)

# Supports de stockage accessoires

## Où sont les disques ?

(question pour les utilisateurs Windows)

Les supports de stockage accessoires sont **montés** sur des **points de montage** (dossiers vides).



Le contenu des **supports de stockage** est dès lors visible comme le **contenu** des **points de montage**.

# /mnt et /media : points de montage

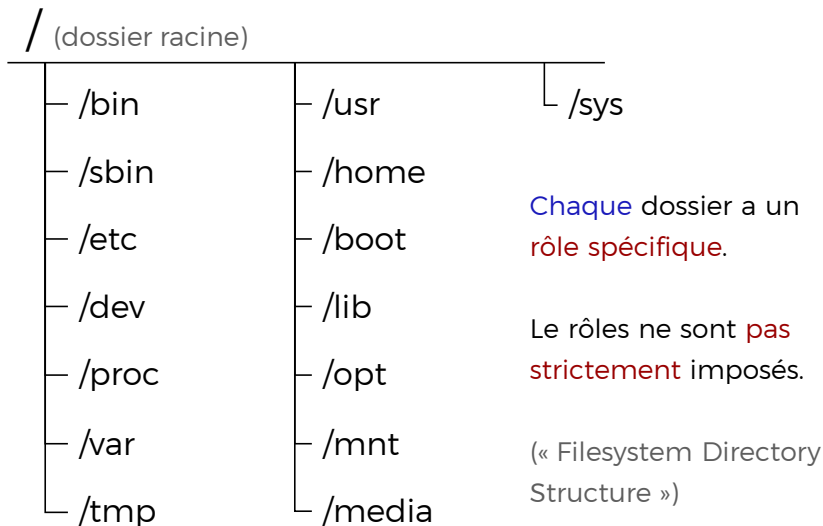
- ▶ contiennent des **points de montage** (des dossiers vides)
- ▶ `/mnt/tmp/`, `/media/cdrom/`

# /mnt et /media : points de montage

- ▶ contiennent des **points de montage** (des dossiers vides)
- ▶ `/mnt/tmp/`, `/media/cdrom/`

La plupart de distributions préfèrent garder les **points de montage** gérés **automatiquement** dans `/media`.

# Rappel : structure typique de dossiers sous Linux



<http://www.thegeekstuff.com/2010/09/linux-file-system-structure/>

# Plan du cours

1. Les dossiers sous Linux

2. Introduction à la ligne de commande



# Répertoire de travail

Le **répertoire de travail** définit votre « localisation » dans l'**arbre de dossiers**.

Tous les chemins qui **ne commencent pas** par **/** sont **relatifs** au **répertoire de travail**.

# Répertoire de travail

Le **répertoire de travail** définit votre « localisation » dans l'**arbre de dossiers**.

Tous les chemins qui **ne commencent pas** par **/** sont **relatifs** au **répertoire de travail**.

`pwd`                      **afficher** le répertoire de travail

# Répertoire de travail

Le **répertoire de travail** définit votre « localisation » dans l'**arbre de dossiers**.

Tous les chemins qui **ne commencent pas** par **/** sont **relatifs** au **répertoire de travail**.

<code>pwd</code>	<b>afficher</b> le répertoire de travail
<code>ls</code>	<b>afficher le contenu</b> du répertoire de travail

# Répertoire de travail

Le **répertoire de travail** définit votre « localisation » dans l'**arbre de dossiers**.

Tous les chemins qui **ne commencent pas** par **/** sont **relatifs** au **répertoire de travail**.

<code>pwd</code>	<b>afficher</b> le répertoire de travail
<code>ls</code>	<b>afficher le contenu</b> du répertoire de travail
<code>cd [dossier]</code>	<b>changer</b> de répertoire de travail pour <b>[dossier]</b>

# Répertoire de travail

Le **répertoire de travail** définit votre « localisation » dans l'**arbre de dossiers**.

Tous les chemins qui **ne commencent pas** par **/** sont **relatifs** au **répertoire de travail**.

<code>pwd</code>	<b>afficher</b> le répertoire de travail
<code>ls</code>	<b>afficher le contenu</b> du répertoire de travail
<code>cd [dossier]</code>	<b>changer</b> de répertoire de travail pour <b>[dossier]</b>

`ls [dossier]` marche aussi pour afficher le contenu de **[dossier]**.

# Création de dossiers et de fichiers

`mkdir [dossier]` créer le `[dossier]`

`mkdir -p [chemin]` créer tous les dossiers dans `[chemin]`

`touch [fichier]` créer le `[fichier]` vide

Tous les chemins qui ne commencent pas par / sont relatifs au répertoire de travail.

# Copier et déplacer des machins

```
cp [source] [destination]
```

copier [source] vers la [destination]

```
cp -r [source] [destination]
```

copier récursivement le répertoire [source] vers la [destination]

```
mv [source] [destination]
```

déplacer [source] vers la [destination]

# Supprimer des trucs

`rm [fichier]`      supprimer le [fichier]

`rm -r [dossier]`    supprimer le [dossier]



# Affichage du contenu texte

cat [fichier]    afficher le contenu du [fichier]

less [fichier]    afficher le contenu du [fichier]  
dans une fenêtre déroulante

# Archivage

```
zip -r [archive.zip] [machin]
```

créer l'archive [archive.zip] à partir de [machin]

```
tar -czf [archive.tar.gz] [machin]
```

créer l'archive [archive.tar.gz] à partir de [machin]

```
unzip [archive.zip]
```

dézipper l'archive [archive.zip]

```
tar -xf [archive.tar.gz]
```

décompresser l'archive [archive.tar.gz]

## Visualiser la documentation

`[commande] --help` demander à `[commande]` d'afficher les consignes d'utilisation

`man [commande]` afficher la documentation système pour `[commande]`

Certains programmes n'installent pas d'entrées dans la documentation système