

Les bases du système linux pour le calcul scientifique

G. Cougoulat / F. Audra

<https://pole-calcul-formation.gricad-pages.univ-grenoble-alpes.fr/ced/>

Séance 1 :

Notions de base sur les systèmes Linux

- ▶ Unix, Linux et les différentes distributions
- ▶ Le système d'exploitation
- ▶ Les différentes formes de terminaux
- ▶ Pourquoi Linux ?
- ▶ Le système de fichiers
- ▶ Arborescence d'un système Linux
- ▶ Les droits d'accès aux fichiers
- ▶ Qu'est ce qu'un shell ?
- ▶ Construire une commande sous Linux
- ▶ Le manuel de référence (man)
- ▶ Les processus
- ▶ Manipuler des processus avec les signaux

Séance 1 : Notions de base sur les systèmes Linux

UNIX, Linux, et les différentes distributions



- ▶ 1969 : Le premier système UNIX est créé par Kenneth Thompson (Laboratoires Bell).
- ▶ 1992 : Linux (GNU/Linux) est une implémentation libre du système Unix.
GNU : projet de système d'exploitation libre initié par Richard Stallman.
Linux : est la contraction de «Linus» et «Unix» du nom de l'initiateur du noyau Linus Torvalds.
- ▶ GNU/Linux est un système multi-tâches, multi-utilisateurs basé sur la famille de normes POSIX (Portable Operating System Interface, X de Unix).
- ▶ Linux est un système Unix. Il est aujourd'hui décliné dans de nombreuses distributions (Debian, Red Hat, ubuntu, Mint, etc...)
- ▶ Chaque distribution propose généralement plusieurs versions d'un même système d'exploitation. Cela peut aller de la version LTS (long time support) très stable et pérenne dans le temps, à des versions plus « expérimentales » et donc beaucoup moins stable (« unstable »).

Séance 1 : Notions de base sur les systèmes Linux

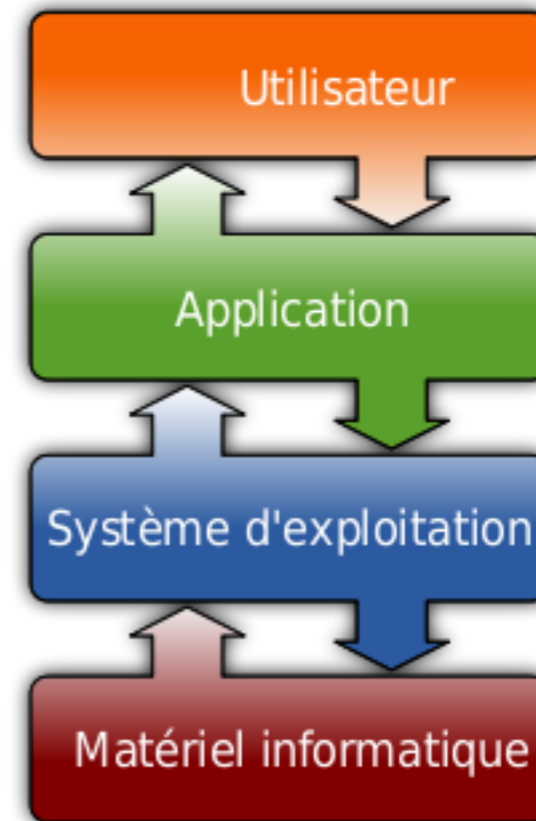
Le système d'exploitation

► Définition :

Le système d'exploitation est l'ensemble des programmes qui font l'interface entre les ressources informatiques matérielles et les logiciels d'applications.

► Exemples :

Windows, Linux, Mac OS X, etc...



Séance 1 : Notions de base sur les systèmes Linux

Les différentes formes de terminaux

- ▶ **Terminal informatique** : périphérique physique de communication entre l'homme et la machine (voir un réseau de machines).
 - de type console : écran / clavier / souris (Terminal X, ...)
 - de type graphiques : environnement fenêtré (utilisation de protocole X-Window et RDP)
 - smartphones, tablettes, liseuses, objets connectés (voitures, TV, réfrigérateurs, etc...)
- ▶ **Terminaux logiciels (émulateur de terminal)** : dit aussi terminaux virtuels, ils simulent le fonctionnement d'une console physique.
- ▶ On appelle communément « terminal » une fenêtre en mode texte à travers laquelle on a accès à un **shell**. Les plus connus de ces terminaux virtuels xterm, konsole, gterm, etc...

Séance 1 : Notions de base sur les systèmes Linux

Pourquoi Linux ?

- ▶ Système d'exploitation OpenSource très répandu dans notre milieu.
Répond aux contraintes du HPC (High Performance Computing) en terme de :
 - Disponibilité (redémarrage très rare de part sa conception modulaire)
 - Pérennité (système basé sur des standards, les normes POSIX)
 - Performances en charge (muti-tâches, et multi-utilisateurs)
 - Stabilité
- ▶ Logithèque très vaste et variée
- ▶ Interface graphique et ligne de commande très puissante
- ▶ La maîtrise d'un système Unix permet de travailler de la même façon sur toutes les plateformes non-Windows.

Séance 1 : Notions de base sur les systèmes Linux

Le système de fichiers

- ▶ Linux a la particularité de considérer la majorité des objets comme des fichiers. Ce concept à l'avantage d'uniformiser un bon nombre d'opérations au niveau du système. **Tout est fichier.**
- ▶ Les périphériques d'entrée/sortie étant considérés comme des fichiers spéciaux, on peut écrire de la même manière sur un disque dur que sur une bande magnétique. On peut lire un fichier de la même façon que l'on peut lire les entrées sur un clavier.
- ▶ Le système de fichier est **hiérarchique**. il est régit par des **droits** et intègre une fonctionnalité de **lien symbolique** (redirection d'un fichier sur un autre).

Séance 1 : Notions de base sur les systèmes Linux

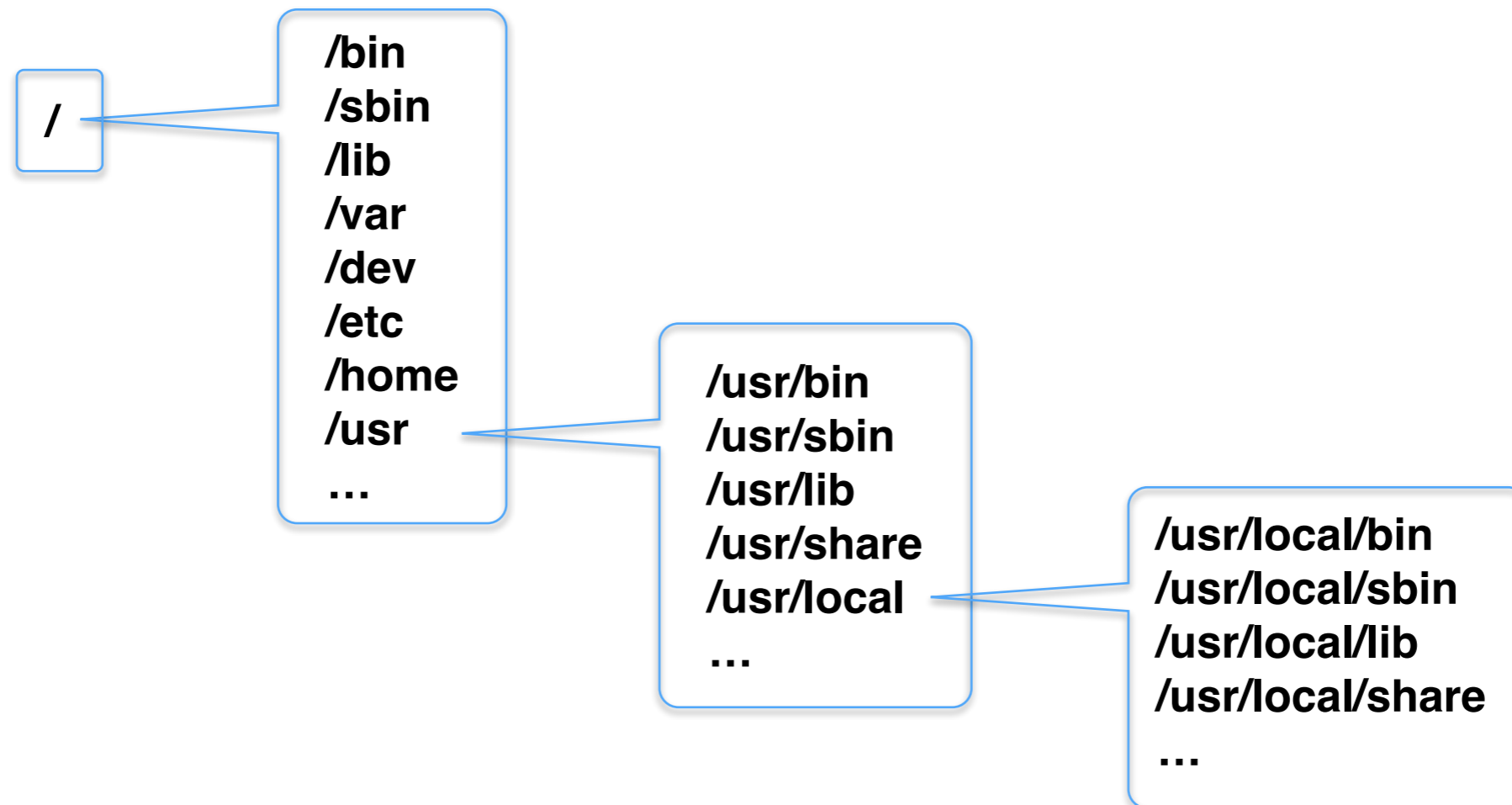
Arborescence d'un système Linux

- ▶ **/** (racine ou root) là où tous les autres répertoires sont montés (accrochés)
- ▶ **/root** répertoire personnel de l'administrateur
- ▶ **/bin** une partie des binaires du système et quelques commandes (ls,cat,rm ..)
- ▶ **/lib** contient les bibliothèque (librairies) nécessaire aux binaires dans /bin
- ▶ **/home** répertoire où sont stockés les répertoires propres aux utilisateurs (/home/toto)
- ▶ **/etc** contient les fichiers de configuration et des fichiers systèmes pour le démarrage
- ▶ **/boot** image du noyau pour Linux
- ▶ **/dev** ensemble des périphériques (devices) (clavier,disques,cartes son et réseau,etc.)
- ▶ **/var** fichiers temporaires de quelques démons,de spools (mail,impression),de logs,de locks
- ▶ **/proc, /sys, /tmp, /mnt**
- ▶ **/opt** lieu d'installation préféré des logiciels "modernes"
- ▶ **/usr** Unix Système Ressources : contient toutes les ressources du système
- ▶ **/usr/bin** pour les binaires liés à une bibliothèque
- ▶ **/usr/lib** pour les bibliothèques nécessaires aux binaires dans /usr/bin
- ▶ **/usr/include** pour les "includes" (les .h)
- ▶ **/usr/share** pour les ressources partagées par les binaires de **/usr/bin**
- ▶ **/usr/sbin** pour les Statics Binaries,les binaires non liés à des bibliothèques
- ▶ **/usr/local** espace «non standard» pour les rajouts de logiciels compilés en local
- ▶ **/usr/local/bin** les binaires compilés en local
- ▶ **/usr/local/lib** les bibliothèques compilées en local
- ▶ **/usr/local/include** pour les "includes" (.h)
- ▶ **/usr/local/src** pour les codes sources

Séance 1 : Notions de base sur les systèmes Linux

Arborescence d'un système Linux

Norme : (File Hierarchy Standard)



Séance 1 : Notions de base sur les systèmes Linux

Les droits d'accès aux fichiers

- ▶ Les fichiers et répertoires d'un système sont régis par des droits organisés de cette façon :

```
d rwx rwx rwx
U  s  G  o
```

- ▶ Visualiser les droits des dossiers et fichiers :

```
jdoue@machine-ciment:~/FormationLinux$ ls -l
total 0
drwxr-xr-x 2 audraf mics 48 12 déc. 14:40 MonDossier/
drwxr-xr-x 2 audraf mics 48 12 déc. 14:40 MonDossier2/
-rw-r--r-- 1 audraf mics  0 12 déc. 14:40 MonFichier
-rw-r--r-- 1 audraf mics  0 12 déc. 14:40 MonFichier2
```

- ▶ Modifier les droits sur les dossiers ou fichiers :

```
jdoue@machine-ciment:~/FormationLinux$ chmod g+w MonFichier
jdoue@machine-ciment:~/FormationLinux$ ls -ld MonFichier
-rw-rw-r-- 1 audraf mics  0 12 déc. 14:40 MonFichier
jdoue@machine-ciment:~/FormationLinux$ chmod o-r MonFichier
jdoue@machine-ciment:~/FormationLinux$ ls -ld MonFichier
-rw-rw---- 1 audraf mics  0 12 déc. 14:40 MonFichier
jdoue@machine-ciment:~/FormationLinux$ chmod u=rwx,go=rx MonFichier
jdoue@machine-ciment:~/FormationLinux$ ls -ld MonFichier
-rwxr-xr-x 1 audraf mics  0 12 déc. 14:40 MonFichier
```

Séance 1 : Notions de base sur les systèmes Linux

Qu'est ce qu'un shell ?

- ▶ Le «shell» est l'interface entre l'utilisateur et le noyau du système Linux
- ▶ Chaque utilisateur est donc connecté sur une machine via un «login shell» propre.
- ▶ Un shell peut lancer des sous-shells (notamment lors d'exécution de scripts)
- ▶ Il existe plusieurs shells (csh, zsh, sh, bash, ...), le plus courant d'entre eux est le «bash».

Séance 1 : Notions de base sur les systèmes Linux

Les différents modes du shell

► Le mode interactif, la ligne de commande et le **prompt** :

```
jdoe@machine-ciment:~$ echo «Bonjour tout le monde !»  
Bonjour tout le monde !
```

► Le mode script :

```
jdoe@machine-ciment:~$ cat MonScript.sh  
#!/bin/bash  
echo «Bonjour tout le monde !»  
  
jdoe@machine-ciment:~$ chmod +x MonScript.sh  
  
jdoe@machine-ciment:~$ ./MonScript.sh  
Bonjour tout le monde !
```

Séance 1 : Notions de base sur les systèmes Linux

Construire une commande sous Linux

► Une commande Unix est composée de 3 parties : le nom de la commande, les options, et les arguments.

► **Syntaxe :**

```
$ commande [options] [arguments]
```

► **Exemple :**

```
$ cd /home/jdoe
```

```
$ ls -l /usr
```

```
$ ps -U jdoe -o ruser,pid,ppid,comm
```

Séance 1 : Notions de base sur les systèmes Linux

Le manuel de référence

- ▶ Le manuel permet d'obtenir de l'aide sur la quasi totalité des commandes. C'est la première commande à apprendre.

```
$ man [options] commande
```

- ▶ On peut également obtenir de l'aide sur les options d'une commande :

```
$ commande --help
```

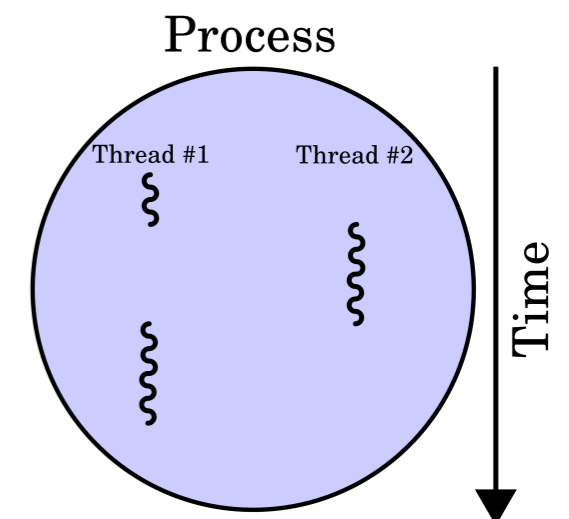
- ▶ Les différentes sections du man :

- 1 Commandes utilisateur
- 2 Appels système
- 3 Fonctions de bibliothèque
- 4 Fichiers spéciaux
- 5 Formats de fichier et protocoles
- 6 Jeux
- 7 Divers
- 8 Administration système
- 9 Interface du noyau Linux

Séance 1 : Notions de base sur les systèmes Linux

Les processus

- ▶ On parle de processus lorsqu'un programme est chargé en mémoire et qu'il s'exécute. Un processus est donc une instance de l'exécution d'un programme en mémoire vive.
- ▶ Sur un système Linux, le processus appartient à l'utilisateur qui l'a lancé. Chaque processus possède un numéro unique sur le système, le PID.
- ▶ Chaque processus a un processus parent que l'on désigne par le PPID. Le processus 1 (init) n'a pas de parent.
- ▶ Un processus peut être multi-threadé.
- ▶ Un processus possède un espace mémoire qui lui est propre, les threads partagent l'espace mémoire du processus dans lequel ils tournent.
- ▶ Un programme parallèle peut être multi-threadé ou multi-processus (communication inter processus) ou les deux à la fois.



Séance 1 : Notions de base sur les systèmes Linux

Manipuler des processus : les signaux

- ▶ Lister les processus du système :

```
jdoe@machine-ciment:~$ ps -ef  
jdoe@machine-ciment:~$ ps --user jdoe --forest -o ruser,pid,ppid,comm
```

- ▶ Visualiser et classer « en direct » les processus :

```
jdoe@machine-ciment:~$ top  
jdoe@machine-ciment:~$ htop
```

- ▶ Lancer un processus en tache de fond :

```
jdoe@machine-ciment:~$ sleep 300 &  
[1] 14948  
  
jdoe@machine-ciment:~$
```


Séance 1 : Notions de base sur les systèmes Linux

Manipuler des processus avec les signaux

► Envoyer des signaux aux processus :

```
# Pour lister les signaux disponibles
```

```
jdooe@machine-ciment:~$ kill -l
```

```
# Envoi le signal par défaut (15 : SIGTERM) au processus de PID = 1225
```

```
jdooe@machine-ciment:~$ kill 1225
```

```
# Tue tous les processus que l'on peut tuer.
```

```
jdooe@machine-ciment:~$ kill -9 -1
```

Séance 1 : Notions de base sur les systèmes Linux

Mémo

Commande	Description rapide
cd	se déplacer dans l'arborescence des fichiers
pwd	affiche le répertoire de travail courant
ls	lister le contenu du répertoire courant
cp	copier un ou plusieurs fichiers
mkdir	créer un répertoire
rm	supprimer un ou des fichiers
mv	déplacer, renommer des fichiers ou des répertoires
man	obtenir de l'aide sur une commande
ln	créer un lien vers un fichier ou répertoire (à utiliser généralement avec l'option -s pour créer un lien « symbolique »)
ps	lister les processus
top ou htop	observer l'évolution des processus sur le système
kill	envoyer un signal à un processus
trap	inhiber ou activer la réaction associée à un signal
cat	concaténe et affiche le contenu de fichiers
less	affiche le contenu d'un fichier pour une lecture plus pratique
find	rechercher un fichier
grep	filtrer un flux
tar	créer une archive à partir d'une arborescence (peut être compressée si besoin)
chmod	changer les droits sur un fichiers