

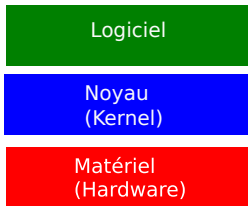
Introduction à GNU/Linux

Miguel Colom, Lara Raad, Aude Bernard-Champmartin

miguel.colom@cmla.ens-cachan.fr, lara.raad@cmla.ens-cachan.fr, champmar@cmla.ens-cachan.fr

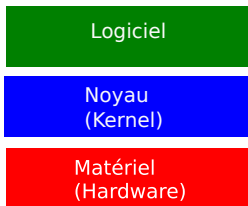
CMLA, ENS Cachan

Septembre 2013



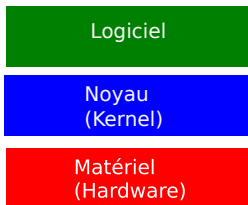
Composantes du système

- Hardware : les composantes électroniques (CPU, mémoire, carte vidéo, carte Ethernet, ...)



Composantes du système

- Noyau (*kernel* en anglais) : intermédiaire entre les logiciels et le hardware. Il gère le fonctionnement du hardware. Exemple : si un programme a besoin d'accéder à la carte vidéo ce n'est pas nécessaire qu'ils connaissent les détails électroniques. Il fait simplement une demande (*system call*) au noyau.



Composantes du système

- Utilitaires. Exemple : LibreOffice, calculatrice, ainsi que les utilitaires qui interprètent les commandes de l'utilisateur, qui gèrent les fichiers, les éditeurs de fichiers, les communications...

Qu'est-ce que Linux ?

Linux

- Linux c'est un noyau. *Juste un noyau.*
- 1991 Linus Torvalds commence la création d'un nouveau noyau qui s'appelle Linux [1].
- Actuellement son développement continue. Exemple : `uname -a`



[1]

<https://groups.google.com/forum/#!topic/comp.os.minix/dINtH7RRrGA%5B1-25-false%5D>

Qu'est-ce que GNU/Linux ?

Système GNU/Linux

- Un système d'exploitation est l'ensemble des logiciels qui permet d'utiliser les ressources d'un ordinateur. Il supervise l'interaction entre le système informatique et un utilisateur de ce système.
- GNU/Linux c'est un noyau + tous les programmes de la distribution, c'est à dire c'est un système d'exploitation complet.

- 1984 Richard M. Stallman commence le projet GNU (GNU is Not Unix, acronyme récursif). Il s'agit de créer un ensemble de programmes *libres* avec les mêmes fonctionnalités que celles de UNIX (qui n'est pas libre).
- UNIX (UNiplexed Informations and Computing Systems) est un système d'exploitation commercial développé à l'origine par Bell et l'université de Berkeley dans les années 70. GNU/Linux et MacOS sont des versions dérivées d'UNIX.



- Initialement le projet GNU n'avait pas de noyau propre.
- Solution : adopter le noyau Linux. C'est le système GNU/Linux.



Qu'est-ce que le logiciel libre ?

Le logiciel libre doit vérifier 4 conditions :

- Liberté de lancer le programme pour n'importe quelle raison
- Liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins
- Liberté de redistribuer des copies
- Liberté d'améliorer le programme et diffuser ces améliorations

La licence GPL

Une des licences de logiciel libre la plus utilisée c'est GPL (GNU Public Licence). [http ://www.gnu.org/licenses/gpl.html](http://www.gnu.org/licenses/gpl.html)

Les points forts du système

- Portable : Linux est écrit à 90 % dans C, un langage indépendant du processeur ; le noyau et les utilitaires sont portables,
- Modulaire : les grandes tâches peuvent être décomposées en modules plus petits
- De maintenance rapide
- Un système ouvert (*open source*) c'est-à-dire dont les sources sont disponibles et accessibles à tous
- Possède un système d'entrée/sortie standardisé et un système de fichiers hiérarchisés avec protection des données (avec permissions)

La gestion des systèmes de fichiers

Principales caractéristiques

- Il existe une unique arborescence de répertoires. Sa racine a pour nom `/`.
- Un système de fichier définit la codification à bas niveau de l'arborescence des répertoires, pour l'écrire dans un dispositif (par exemple, un disque dur).
- Etant donné que l'arbre d'arborescence est unique il y a une opération qui s'appelle **mount** et qui permet de "greffer" un système de fichier externe dans l'arbre d'arborescence.
- Exercice : Insérer une mémoire USB et visualiser où dans l'arborescence elle a été "greffée" (montée).

L'accès aux fichiers

- Pour accéder à un fichier particulier, outre son nom, il faut indiquer l'ensemble du **chemin** d'accès (**pathname**) à partir de la racine /
- Il existe la notion de **répertoire courant** ou de **travail** (**current directory**). Les références aux fichiers se font alors par rapport à la position de ce répertoire (on n'utilise plus dans ce cas en début de chemin le symbole /)
- Unix et GNU/Linux maintient la trace du **répertoire d'accueil** de chaque utilisateur (**home directory**). Tout utilisateur se trouve dans son répertoire d'accueil lorsqu'il se connecte.
- **Remarque** : Il existe deux noms symboliques :
 - le "." : représente le répertoire lui-même
 - le ".." : représente le répertoire parent.

Environnement de bureau

KDE

- L'environnement de bureau est un logiciel qui permet à l'utilisateur d'interagir avec le système à l'aide d'un environnement graphique.
- Un des plus utilisés c'est KDE. Il en existe d'autre comme GNOME, Unity, ...
- 1996 Matthias Ettrich lance le projet KDE pour offrir aux utilisateurs des systèmes du type Unix une interface graphique unifiée.
- Dolphin est le gestionnaire de fichier par défaut de KDE.
- Exercice : ouvrir Dolphin et naviguer.

La ligne de commandes

- C'est une façon d'exécuter des programmes sans utiliser d'environnement graphique
- Le programme qui permet de faire cela s'appelle *shell*, *console*, *terminal*...
- Le shell permet d'écrire une instruction, l'exécuter et voir sa sortie
- Il est possible de lancer un ou plusieurs shell dans l'environnement graphique

Exercice :

Ouvrir un terminal et exécuter l'ordre **cal** pour visualiser le calendrier

Les avantages d'utiliser un terminal

Quelques avantages :

- Rapidité : possibilité de réaliser un tâche complexe en une seule ligne au lieu de plusieurs clics et mouvement de souris
- Travailler à distance
- Programmable : possibilité d'exécuter des *scripts*

Exercice :

- 1 Créer un fichier de texte avec votre éditeur préféré et écrire quelques lignes. Sauvegarder dans votre dossier personnel avec le nom *exemple.txt*
- 2 Ouvrir un terminal
- 3 Exécuter **man wc** pour obtenir des informations sur cette instruction. Cet ordre permet de compter le nombre de mots, de lignes, ...
- 4 Exécuter **wc -l exemple.txt** pour voir le nombre de lignes dans le fichier
- 5 Exécuter **wc -w exemple.txt** pour voir le nombre de mots dans le fichier

L'utilisateur

Les utilisateurs

GNU/Linux est système multi-utilisateur.

Exercice : exécuter la commande **whoami**

Chaque fichier/répertoire à un propriétaire

Exercice : exécuter **ls -l**

Les groupes

Les utilisateurs peuvent appartenir à un ou plusieurs groupes

Exercice : exécuter la commande **groups**

Les permissions

- Chaque fichier a des permissions différentes pour son propriétaire, son groupe et pour le reste des utilisateurs
- Les permissions sont : lecture (r), écriture (w) et exécution (x). Les permissions peuvent être combinées.
- Pour un dossier x c'est la permission pour rentrer. (Si c'est un fichier c'est la permission d'exécution)

```
total 400
drwxrwxr-x 2 lara lara 4096 août 27 17:25 images
-rw-rw-r-- 1 lara lara 3498 août 30 17:55 slides.aux
-rw-rw-r-- 1 lara lara 157292 août 28 16:46 slides.dvi
-rw-rw-r-- 1 lara lara 46358 août 30 17:55 slides.log
-rw-rw-r-- 1 lara lara 1884 août 30 17:55 slides.nav
-rw-rw-r-- 1 lara lara 163 août 30 17:55 slides.out
-rw-rw-r-- 1 lara lara 160135 août 30 17:55 slides.pdf
-rw-r--r-- 1 lara lara 0 août 30 17:55 slides.snm
-rw-r--r-- 1 lara lara 6753 août 30 17:55 slides.tex
-rw-r--r-- 1 lara lara 2508 août 26 17:40 slides.tex~
-rw-rw-r-- 1 lara lara 249 août 30 17:55 slides.toc
-rw-r--r-- 1 lara lara 1045 août 26 17:26 texput.log
lara@lara-HP-Pavilion-dm1-Notebook-PC:~/Dropbox/BACKUP/Mission supplementaire/2013-2014/GNU-Linux/latex_y_linux/presentation$
```

Exercice

Exécuter **ls -l /etc/magic**

Ouvrir ce fichier avec un éditeur de texte

Essayer de le modifier et de le sauvegarder. Qu'est ce que vous observez ?

Pourquoi ?

Changer les permissions : la commande chmod

La commande chmod

- Permet de changer les permissions de l'utilisateur, groupe, etc.
- Visualiser la description en exécutant **man chmod**

Exemple 1

- 1 Exécuter la commande **echo Hello world > fichier.txt**. La commande **echo** a comme sortie son paramètre. Le symbole **>** redirectionne à un fichier.
- 2 Exécuter **ls -la fichier.txt** pour visualiser les permissions du fichier.
- 3 Qui est le propriétaire de ce fichier ?
- 4 Qu'est ce que peut faire le propriétaire avec ce fichier ?
- 5 Est-ce que le propriétaire peut lire le fichier ?
- 6 Exécuter **cat fichier.txt** pour visualiser son contenu.
- 7 Exécuter **chmod u-r fichier.txt** pour éliminer (-) la permission de lecture (r) du propriétaire (u).
- 8 Exécuter **ls -la fichier.txt** pour visualiser les permissions du fichier.
- 9 Exécuter **cat fichier.txt** pour visualiser son contenu.
- 10 Exécuter **chmod u+r fichier.txt** pour donner (+) la permission de lecture (r) du propriétaire (u).
- 11 Exécuter **cat fichier.txt** pour visualiser son contenu.

Exemple 2

- 1 Exécuter la commande **mkdir reper**. La commande **mkdir** permet de créer un répertoire
- 2 Exécuter **cd reper** pour rentrer dans le répertoire.
- 3 Exécuter **pwd** (Print Working Directory)
- 4 Exécuter **cd ..** pour sortir du répertoire.
- 5 Exécuter **pwd**
- 6 Eliminer la permission du propriétaire d'entrer au répertoire
- 7 Rentrer dans le répertoire
- 8 Y êtes-vous parvenu ? Sinon que devez vous faire pour pouvoir rentrer ?
- 9 Rentrer dans le répertoire

Compiler et exécuter un programme en C

Editer le code source

Créer un fichier code.c avec ce contenu :

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, const char * argv[]) {
    printf("Hello World\n");
    return EXIT_SUCCESS;
}
```

Compiler et exécuter

- 1 Compiler le code source en exécutant **gcc code.c -o code** et l'exécuter avec **./code**
- 2 Visualiser les permissions de ce fichier avec **ls -l code**
- 3 Eliminer le permis d'exécution puis essayer de l'exécuter
- 4 Remettre le permis d'exécution puis exécuter

Quelques commandes

man

Pour voir la référence d'une commande.

Exemple : **man chmod** pour obtenir des informations de la commande chmod.

echo

Affiche la chaîne de caractères passée en paramètre.

Exemple : **echo Hello world** pour afficher Hello world sur l'écran.

cd

Change Directory.

Exemple : **cd ..** pour aller au répertoire précédent.

pwd

Print Working Directory.

Exemple : **pwd**.

ls

LiSte le contenu d'un répertoire.

Exemple : **ls -lh** liste le contenu avec ses détails (-l) en utilisant les suffixes pour Byte, Kilobyte, Megabyte, ...(-h).

cat

Lit le contenu du fichier et l'affiche.

Exemple : **cat code.c**.

cp

CoPy.

Exemple : **cp code.c reper/** copie le fichier code.c dans le répertoire reper.

chmod

Change les permissions du fichier.

Exemple : **chmod g-x code** élimine le permis d'exécution des utilisateurs qui appartiennent au groupe.

rm

ReMove.

Exemple : **rm reper/code.c**.

mkdir

MaKe DIRectory.

Exemple : **mkdir -p rouge/vert/bleu** crée le répertoire rouge dans l'actuel puis vert dans rouge et bleu dans vert.

mv

Déplace ou change le nom d'un fichier ou répertoire.

Exemple : **mv code.c programme.c** change le nom du fichier code.c a programme.c.

df

Disk Free. Pour visualiser l'espace disponible et occupé du disque.

Exemple : **df**.

date

Afficher ou modifier la date.

Exemple : **date "+%Y"** . Pour visualiser l'année actuelle.

touch

Modifie la date de la dernière modification du fichier et la date du dernier accès. Si le fichier n'existe pas il est créé.

Exemple : **touch nouveau_fichier.txt**.

wget

Web Get. Télécharge un fichier à partir de l'URL indiquée.

Exemple : **wget -c**

http://www.gnu.org/graphics/gerwinski-gnu-head.png

Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

