

Fondamentaux de l'informatique : Linux

03/10/2018

Catherine ATTAL

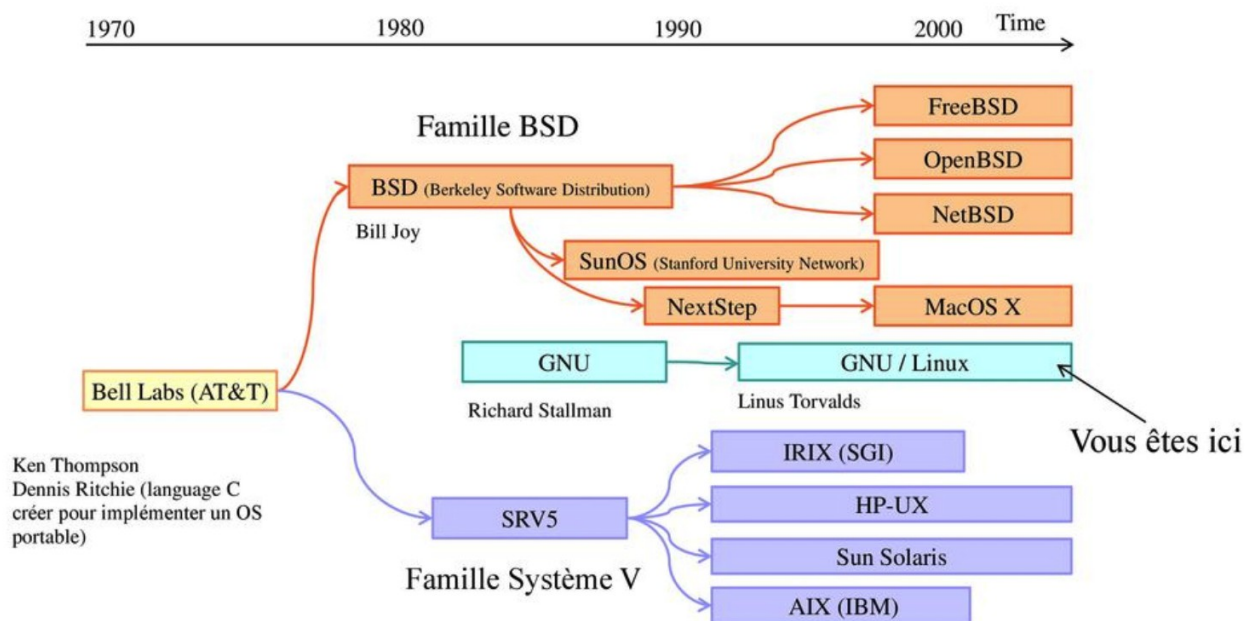
cattal@pierrefitte93.fr

Slides du cours : <https://slideplayer.fr/slide/11998952/>

Introduction à Unix et GNU

Histoire de Linux

Arbre généalogique d'Unix



L'histoire de l'informatique commence probablement avec les bouliers : l'informatique n'étant avant tout qu'une méthode pour stocker et traiter de l'information. L'histoire de Linux quant à elle commence en 1970 avec l'entreprise Bell Labs qui conçoit une méthode en langage C pour pouvoir dialoguer avec une machine pour lui donner des ordres. La machine fonctionne en binaire : c'est extrêmement limité, soit il y a un flux, soit il y en a pas. Le langage C permet d'implémenter un système d'exploitation portable sur la machine.

Par la suite, Bell Labs fait faillite et le code source est mis à disposition de son langage appelé Unix est mis à disposition. Des entreprises se l'approprient et deux grandes familles se créent : BSD et SVR5 qui correspondent à deux sous-familles du code source Unix. Ces deux branches sont des branches propriétaires : les codes sont fermés et on a pas le droit de les modifier.

BSD engendre *in fine* MacOS X. La famille SRV5 donne des systèmes d'exploitation pour des très gros réseaux, type IBM : ce sont des gros systèmes hyper solides avec une sécurité très avancée.

GNU est créée en 1983 par Richard Stallman : c'est un système non Unix en réaction justement un

systèmes propriétaires des deux autres branches. Il veut un code qui appartienne à la collectivité. Il développe un cœur de système qu'il nomme Linux. Aujourd'hui on utilise des dérivés Linux comme Ubuntu.

Philosophie d'Unix et ses caractéristiques

Le principe fondamental qui régit le code de Unix est ce qui est simple est beau : on vise la façon de programmer la plus épurée possible. Les programmes sont chargés de faire une seule chose et de la faire bien. On essaie également d'éviter les interfaces captives : les programmes sont censés pouvoir tourner sur n'importe quelle machine ou système.

Le système se compose de plusieurs niveau :

- Le noyau : le niveau le plus bas et le moins subtil, c'est le niveau **matériel**.
- Le shell (coquille) : une interface pour permettre à l'utilisateur de communiquer avec le noyau, c'est le niveau **texte**.
- X Window : habillage visuel du tout, c'est le niveau **graphique**.

Ce qui distingue Unix au départ c'est que c'est un système multi-utilisateur, c'est-à-dire qu'il peut y avoir plusieurs profils sur un même poste et chacun fonctionnera indépendamment. Il peut traiter en simultané plusieurs demandes. Le système admet également un utilisateur root qui a tout les droits sur le système : c'est une sorte de super-administrateur qui peut aller jusqu'à supprimer le système lui-même. Les droits root sont donc à manier avec précaution.

Ce qui fait la spécificité d'Unix c'est que le système est également multitâches : plusieurs programmes peuvent tourner en même temps (traitement de texte et calcul par exemple). Il supporte de plus plusieurs processeurs (donne la puissance de calcul à l'ordinateur, traite les *process* qui sont des trains de bits) et prend en charge le réseau nativement : c'est en somme un système extrêmement flexible.

Les notions fondamentales d'Unix sont :

- La **portabilité** : peut être exécuté sur un grand nombre de machines différentes (que ce soit Mac ou Microsoft) car il va directement dialoguer avec la machine (et n'interagit pas avec les couches graphiques)
- La **scalabilité** : il n'est pas figé dans le temps et tout le monde peut le faire évoluer. On peut créer son propre système.

Les différentes couches d'un système Unix

Architecture d'un système d'Unix

Applications graphiques des utilisateurs	Navigateur web, traitement de texte, multimédia	ESPACE UTILISATEUR
Application en ligne de commande	Terminal : ls, mkdir, wget, ssh, gcc, busybox, etc.	
Librairies partagées	Driver d'imprimantes, libjpeg, libstdc++, libxml, etc.	
Librairie C	GNU C library, UClibc	
Noyau système	Linux, Hurd	ESPACE NOYAU
Matériel et périphérique	CD, imprimante, ordinateur	MATÉRIEL

Pour contrôler les éléments les plus bas, il faut installer des drivers qui se stockent dans les différentes couches : la couche noyau parle directement au disque dur en binaire, ensuite les librairies, utilisent plusieurs sortes de codes pour faire faire différentes actions au noyau. Il existe des

librairies de basse couche comme de haute couche pour agir sur les différents niveaux du système.

Les applications de ligne de commandes comme le terminal communiquent avec les librairies qui elles-mêmes communiquent avec le noyau. S'il y a un problème, il faut toujours commencer par voir si le problème ne vient pas des couches les plus basses (en commençant par le matériel) en remontant vers les couches les plus hautes. Ubuntu correspond à l'application graphique c'est-à-dire la couche le plus haut niveau là où Linux correspond à l'espace noyau. Tout ce qui peut être fait dans la couche graphique peut être fait en ligne de commande sur le terminal sans interface graphique.

Le projet GNU, licence de logiciels libres

Le projet GNU

GNU signifie : **GNU** is **N**ot **U**nix (acronyme récursif).

Le projet de Stallman était de faire un système entièrement libre qui puisse continuer à évoluer après lui. Au départ, le projet s'est beaucoup développé dans les universités. Les composants initiaux de GNU sont un compilateur C (gcc), make (GNU make), Emacs, bibliothèque C (glibc), outils de base (ls, cp ...). Cependant en 1991, le projet GNU n'avait toujours pas de noyau et tournait sur des Unix propriétaires.

Les logiciels libres

La licence logiciel libre garantit 4 libertés :

- liberté d'utiliser le logiciel à n'importe quelle fin,
- liberté de le modifier (les sources doivent être en libre accès et que le programme soit décrit dans des fichiers mis à disposition),
- liberté de redistribuer des copies
- liberté et partager les modifications qu'on a faites.

Linux est sous licence BSD : elle garantit ces 4 libertés cependant elle permet de rendre propriétaire le programme pour soi. On peut payer la licence. C'est le cas de Apache par exemple. GPL signifie *General Public Licence*. La GNU GPL est la plus répandue : cette licence exige que toutes les versions améliorées publiées soient du logiciel libre.

Les logiciels libres se distinguent des gratuits : ces logiciels ne sont pas forcément gratuits car il peuvent être sur une licence payante. Ils participent cependant d'un mouvement basé sur la liberté individuelle et l'utilité sociale de la coopération. L'*open source* est différent car c'est une approche plus pragmatique : les sources sont publiées mais il n'y a pas forcément le droit de faire ce qu'on veut avec.

Linux, Distributions GNU/Linux

Linux

Linux a été conçu par Linus Torvalds en 1991. Linux repose sur les outils GNU : bibliothèque C, gcc, binutils, make, emacs, etc. Le système est appelé GNU/Linux. Il est très tôt partagé sous licence GPL et les utilisateurs et contributeurs sont de plus en plus nombreux : il connaît ensuite une croissance supérieure à tout autre système d'exploitation.





Distribution GNU/Linux

Les distributions sont des compilations, c'est-à-dire des packages tout prêts où sont toutes les briques nécessaires à l'utilisation d'un système ou programme. Ces distributions sont disponibles sous forme de paquetages qui sont ce qu'on installe pour faire fonctionner quelque chose. Paquet XML par exemple. Ne pas confondre la version de distribution avec celle du noyau.

Distributions commerciales

- Red Hat: <http://www.redhat.com/>
La plus populaire. Fiable, sûre, conviviale et facile à installer, prise en charge par tous les fournisseurs de logiciel et de matériel. 
- Suse (Novell): <http://www.suse.com/>
L'alternative principale. Facile à installer, conviviale et stable. Obtiens le support des fournisseurs de logiciel et de matériel.. 
- Mandriva (anciennement Mandrake): <http://mandrivalinux.com/>
Conviviale, facile à installer, plus innovante, mais moins stable. Cible principalement les utilisateurs individuels. Peu pris en charge par les fournisseurs de logiciel et de matériel. 

Distributions communautaires

- Fedora Core: <http://fedora.redhat.com/>
Stable, sûre, conviviale, facile à installer. Sortie fréquente de nouvelles versions complètes. 
- Ubuntu Linux: <http://ubuntu-linux.org/>
La distribution communautaire qui progresse le plus.
Basé sur Debian mais avec une version stable tout les 6 mois.
Conviviale pour les utilisateurs. Bonne pour les débutants. 
- Debian: <http://debian.org/>
Très stable et sûre, mais plus difficile à configurer et à installer. Conviviale pour les développeurs mais pas encore pour les utilisateurs. Version stables pas assez fréquentes (tous les 2 ou 3 ans). La meilleure pour les serveurs, mais pas pour les débutants. 
- Mandriva Community: <http://mandrivalinux.com/>
Facile à installer, sûre, conviviale, sortie fréquente de versions complètes, mais moins stable (pas assez de tests et de prise en compte des retours des utilisateurs et testeurs). 

Distributions live

Les distributions live sont des logiciels complets : la forme live correspond à son stockage sur un périphérique (souvent USB) qu'on connecte à un ordinateur et qui boot dessus. Cela permet de réparer un ordinateur ou de récupérer des fichiers. Ça permet aussi de démarrer sur Linux avec n'importe quel ordinateur. Rien n'est installé sur le disque.

Les versions Knoppix sont des distributions live très utilisées dans les écoles pour jeunes enfants. Elles s'installent vraiment rapidement. En 2004 sort Ubuntu. La dernière version stable est 18.04.1 LTS (Long Term Support). Une version non stable continue à évoluer en attendant une autre version stable. 4 et 1 correspondent aux versions du noyau.

Shells, interpréteur de commandes et interpréteur de fichiers

Tout est fichier

Dans Unix tout est un fichier. N'importe quel objet est décrit dans un fichier : une image peut être ainsi modifiée dans sa description fichier. On dénombre comme type de fichier :

- Les fichiers ordinaires
- Les répertoires : des fichiers listant plusieurs fichiers. Un dossier correspond donc à un fichier qui contient la liste de tous les fichiers contenus dans le dossier.
- Les liens symboliques : ils permettent à plusieurs noms de fichiers de faire référence au même contenu sur le disque. Ce sont des sortes de synonymes d'aliases. Utile dans le cadre de publication web : dans un répertoire, on peut trouver un lien vers un fichier dans un autre répertoire sans qu'on en sache l'emplacement.
- Les périphériques et dispositifs : la lecture et l'écriture à partir d'un dispositif de fait comme un fichier
- Pipes : utile pour mettre en cascade plusieurs programmes. Permet de dire fais ça puis après fais ça. Cumuler des commandes en utilisant le caractère |. On peut ainsi décrire quelque chose qu'on a envie de faire.
*Ex : cat *.log | grep error*
- Sockets : communication inter-processus. Une erreur de socket correspond à un problème de communication entre les différentes parties nécessaire à l'exécution d'une commande.

Noms de fichiers

Linux est sensible à la casse : un nom de fichier en majuscule n'est pas équivalent à un en minuscule. Il n'y a également pas de longueur limite imposée, un nom de fichier peut être super long. Les noms peuvent contenir tous les caractères à l'exception de /. Les extensions ne sont pas obligatoires car Linux va chercher directement dans les métadonnées du document pour l'ouvrir avec le logiciel adapté.

Structure des systèmes de fichiers GNU/Linux

Chemin de fichier (path)

Un chemin est la manière d'aller d'un point à un autre. C'est une séquence de répertoires emboîtés avec un fichier ou répertoire à la fin. Pour décrire un *path* absolu, on part toujours de root, c'est-à-dire la base du système : il commence toujours donc par « / » ou par « ./ » qui correspond au niveau root.

Le chemin relatif est relatif au répertoire courant de l'application, c'est-à-dire relatif par rapport à l'endroit où on se trouve. Le chemin absolu part de root. Un chemin relatif ne commence pas par un backslash.

Structure des fichiers dans GNU/Linux

/	Répertoire racine
/bin/	Commandes de base du système
/boot/	Images, initrd et fichiers de configuration du noyau
/dev/	Fichiers représentant des périphériques /dev/hda: premier disque dur IDE
/etc/	Fichiers de configuration du système
/home/	Répertoires utilisateur
/lib/	Bibliothèques de base du système (partagées)

Il s'agit des noms des sous-répertoires du système GNU/Linux. Les noms peuvent varier d'un système à l'autre.

Interpréteurs de commandes

Il est possible d'ouvrir plusieurs pages ou onglet de terminal pour effectuer des actions à différents endroits du système ou encore tester des commandes et les exécuter de manière sécurisée. Linux de manière générale est sensible à la casse, ce qui implique comme bonne pratique de toujours respecter les même règles de nomenclature pour les dossiers et fichiers.

Le terminal présente l'avantage de différencier par les couleurs différents type d'objets. Le bleu représentera les dossier là où le vert correspondra à une information. La première ligne affichée indique quel utilisateur est connecté : `segouy@segouy-VirtualBox:~$` il s'agit ici de segouy.

Entre chaque commande, il est nécessaire de mettre un espace.

<code>pwd</code>	Indique l'endroit où l'on se trouve dans le terminal
<code>ls</code>	Liste le contenu du dossier où l'on se trouve
<code>ls -al</code>	Liste de manière détaillée le contenu du dossier

```

drwx-----+ 17 segolene staff    544 10 oct 18:42 Desktop
drwx-----+  6 segolene staff    192 12 sep 21:24 Documents
drwx-----+  7 segolene staff    224  9 oct 13:01 Downloads
drwx-----@ 72 segolene staff   2304  8 oct 13:22 Library
drwx-----+  4 segolene staff    128 14 jan  2018 Movies
drwx-----+  5 segolene staff    160 19 jan  2017 Music
drwx-----+  6 segolene staff    192  5 avr  2017 Pictures
drwxr-xr-x+  5 segolene staff    160 21 jan  2016 Public

```

La première lettre (ou *digit*) sur la gauche correspond à s'il s'agit d'un dossier (cette lettre est alors un `d`) ou d'autre chose (le *digit* est un `-`). Les trois *digits* suivants correspondent aux droits de l'utilisateur principal sur le dossier ou fichier : `r` correspond au droit de lire, `w` à celui d'écrire et enfin `x` à celui d'exécuter. Les trois *digits* suivants correspondent aux droits du groupe et enfin les derniers à ceux d'un utilisateur non-identifié.

La colonne suivante affiche le nom de l'utilisateur et celle à sa droite affiche le groupe auquel appartient l'utilisateur. La colonne de chiffres à droite correspond à la taille du dossier ou document (pas sur) et les suivantes donnent des informations sur la date et l'heure de la création. Le nom du fichier/dossier apparaît à droite. Les points correspondent au niveau de l'arborescence (pas sur d'avoir compris)

<code>cd nom_dossier</code>	Aller dans le dossier ci-nommé lorsqu'on se trouve dans le dossier parent
<code>cd ..</code>	Remonter au niveau supérieur de l'arborescence
<code>clear</code>	Nettoyer l'écran
<code>mkdir Chartes</code>	Créer un dossier « Chartes » là où on se trouve (<i>make directory</i>)
<code>rmdir Chartes</code>	Supprime le dossier « Chartes » s'il est vide (<i>remove directory</i>)
<code>mv Chartes M2</code>	Renommer le dossier « Chartes » en « M2 »

L'autocomplétion permet de compléter automatiquement le nom d'un dossier/fichier avec les premières lettres en appuyant sur `tab`. Si plusieurs objets commencent par la même lettre, appuyer deux fois sur `tab` permet d'afficher ces différents éléments. Les flèches du haut et du bas permettent de parcourir les précédentes commandes sans avoir à les taper à nouveau.

<code>vi fichier</code>	Créer un fichier de traitement de texte Vi qui s'appelle « fichier » => ouvre automatiquement le fichier au sein du terminal. Les flèches (et le scroll) font se déplacer en inscrivant les lettres A B C et D si la touche insert est activée
<code>echap+:w</code>	Enregistrer le fichier
<code>echap+:q</code>	Quitter et revenir au terminal
<code>dd</code>	Supprimer des caractères
<code>vi fichier</code>	Ouvrir à nouveau le fichier qui a été créé

`less mon_fichier`: voir le contenu du fichier « mon_fichier » sans avoir être dessus ou à l'éditer ensuite `echap + : + q`

La commande ls

<code>ls -a</code>	Affiche tous les fichiers y compris les fichiers cachés
<code>ls -t</code>	Affiche les fichiers les plus récents en premiers
<code>ls -S</code>	Affiche les fichiers les plus gros en premiers
<code>ls -l</code>	Affiche tous les détails des fichiers
<code>ls -r</code>	Affiche en ordre inversé
<code>ls -ltr</code>	Affiche tous les fichiers, les plus récents en premier en ordre inversé
<code>ls *txt</code>	Affiche tous les fichiers qui finissent par txt
<code>ls D*</code>	Affiche tous les fichiers qui commencent par D
<code>ls -d ex*</code>	Affiche tout – y compris les dossiers – qui commencent par ex

La commande cat

<code>cat?.log</code>	Affiche le contenu de tous les fichiers dont le nom commence par un caractère et finit par .log
<code>cat</code>	Permet de rechercher à l'intérieur des fichiers

c'est pas bien clair

Les copies : la commande cp

<code>cp toto tutu</code>	Copier le fichier « toto » et le renommer en « tutu » deux fichiers identiques nommés différemment.
<code>cp tutu coucou/tutu2</code>	Copier « tutu » dans le dossier « coucou » et le nommer « tutu2 »
<code>cp tutu home/utilisateur</code>	Copier « tutu » dans le dossier « utilisateur », dossier parent au nôtre
<code>rm tutu</code>	Supprimer le fichier « tutu »
<code>cp -R linux /home/segouy/Chartes</code>	Copier le dossier « linux » dans « Chartes »
<code>cp fich1 fich2 fich3 répertoire</code>	Copier plusieurs fichiers dans le dossier « répertoire »

La commande grep

Cette commande permet de peaufiner la recherche ou le listage

`ls -al |grep D*` : liste tout ce qui commence par D dans le dossier de manière détaillée

- ▶ **grep <motif> <fichiers>**
Parcourt les fichiers donnés et affiche les lignes qui correspondent au motif spécifié.
- ▶ **grep erreur *.log**
Affiche toutes les lignes contenant **erreur** dans les fichiers ***.log**
- ▶ **grep -i erreur *.log**
Idem, mais indifférent aux majuscules / minuscules
- ▶ **grep -ri erreur .**
Idem, mais récursivement dans **.** et ses sous-répertoires
- ▶ **grep -v info *.log**
Affiche toutes les lignes des fichiers, sauf celles qui contiennent **info**

Droits d'accès en octal

Avec la commande `chmod` on peut modifier les droits d'accès à un fichier ou dossier. Attention à ne pas se supprimer les droits d'accès en indiquant 0 au premier chiffre. 777 correspond aux droits les plus permissif et 000 à absolument aucun droit d'accès pour qui que ce soit.

Correspondances de représentation des droits		
Droit	Valeur alphanumérique	Valeur octale
aucun droit	---	0
exécution seulement	--x	1
écriture seulement	-w-	2
écriture et exécution	-wx	3
lecture seulement	r--	4
lecture et exécution	r-x	5
lecture et écriture	rw-	6
tous les droits (lecture, écriture et exécution)	rwX	7

<code>chmod</code>	Changer les permissions à un fichier ou un dossier
<code>Chmod 777 tutu</code>	Donner tous les droits à toute sorte d'utilisateur sur le fichier « tutu »

Interpréteur de fichiers et répertoires

Afficher, trier et scanner un fichier

Lien symbolique et physique

Droit d'accès aux fichiers

Entrée/Sortie standards, redirections, pipes

Entrée et sortie standard, redirections

Pipes : redirection de la sortie standard à une autre commande

Divers

Éditeurs de texte

Compression et archivage

Impression

Comparer des fichiers et des répertoires

Recherche de fichiers

Récupérer de informations sur les utilisateurs