

L'administration d'un serveur Apache sous Linux

Module 1 - Introduction à Linux



Objectifs

- Découvrir le système d'exploitation GNU/Linux
- Apprendre à utiliser le shell

D'Unix à Linux

- **1965** : Bell Labs travaille avec d'autres institutions au développement de **Multics**
- **1969** : Bell Labs se retire de **Multics**
- **1970** : Ken Thompson et Dennis Ritchie sortent **PDP-7**, première version d'Unix
- **1972** : Dennis Ritchie et Brian Kernighan développent le **langage C**
- **1973** : **réécriture** d'Unix en C

D'Unix à Linux

- **1979** : industrialisation et commercialisation d'Unix
- **1979** : développement de la première version de **BSD** par l'université de Berkeley
- **1982** : volonté de convergence et de mettre en place des standards par l'IEEE : standard **POSIX**
- **1991** : premières machines 32 bits personnelles et première version du **noyau Linux**

Introduction à Linux

D'Unix à Linux

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: comp.os.minix

Subject: What would you like to see most in minix?

Summary: small poll for my new operating system

Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>

Date: 25 Aug 91 20:57:08 GMT

Organization: University of Helsinki

Hello everybody out there using minix –

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them 😊

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes – it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

Introduction à Linux

D'Unix à Linux

- **1991-1994** : développement du noyau Linux, création des premières communautés
- **1995** : premières distributions Linux et sociétés commerciales (Red Hat)
- **1996** : début de reconnaissance auprès du public
Premières interfaces bureautiques : KDE
- **1997-1999** : intégration de Linux dans les clusters, l'embarqué et les entreprises
Premiers salons commerciaux

D'Unix à Linux

- **2002** : intégration officielle dans les serveurs HP et IBM
- **2004** : première distribution grand public [Ubuntu](#)
- Depuis, Linux ne cesse de s'étendre dans tous les contextes et devient un standard à la place des autres systèmes Unix
- **2016** : Microsoft annonce que [SQL Server](#) tourne sous Linux

Les distributions

- Une **distribution** Linux est un **ensemble cohérent** de logiciels, certains du projet GNU, assemblés autour du noyau Linux
- Il existe plusieurs centaines de distributions Linux :
<http://distrowatch.org/dwres.php?resource=popularity>
- Certaines sont généralistes (Debian, Ubuntu, Fedora...), d'autres sont dédiées à des utilisations particulières (Kali Linux, Tails...)

Les distributions

- Pour l'ensemble de ce cours, c'est la distribution Debian qui sera utilisée. C'est une distribution souvent conseillée dans les contextes de serveurs de production.

<https://www.debian.org/>



Les licences

- Le **projet GNU** est initié par Richard Matthew Stallman (RMS) en 1984
- Réécriture complète sous licence libre d'un système d'exploitation Unix
- Gros problèmes pour le développement du noyau Hurd
- Le noyau Linux s'intègre avec le projet GNU, d'où l'appellation **GNU/Linux**

Les licences

- La **Free Software Foundation** est créée en 1985 par RMS pour aider au financement des logiciels libres
- La licence la plus utilisée est la **GPL** (*General Public License*)
La version actuelle est la GPLv3
- Une **licence libre** permet de protéger un logiciel pour éviter qu'une société y mette un copyright

Les licences

- Un logiciel libre garantit les libertés suivantes :
 - La **liberté d'exécuter** le logiciel quel que soit le contexte
 - La **liberté d'étudier le fonctionnement** du logiciel, de l'adapter au(x) besoin(s), et donc l'obligation d'accéder au code source
 - La **liberté de redistribuer** des copies
 - L'**obligation de faire bénéficier la communauté** des modifications

Les licences

- Il existe d'autres licences, telles que BSD, CeCILL, Creative Commons...

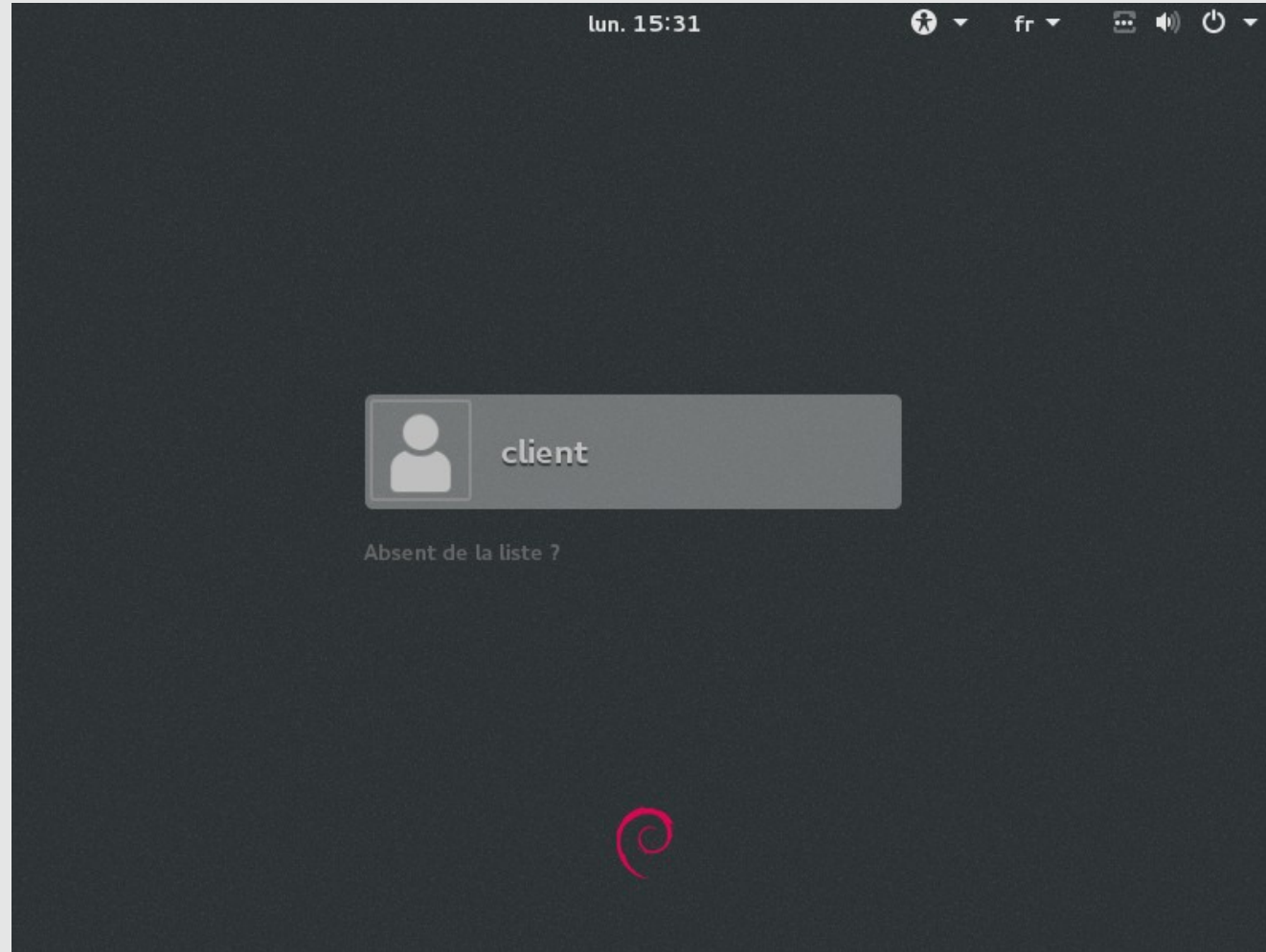
Voir <https://vvlibri.org/fr>

Se connecter

- La première étape pour commencer à travailler sur un système GNU/Linux est la **connexion**
- Elle se fera soit au travers d'une interface graphique pour les environnements bureautiques, soit sous forme de prompt demandant un login et mot de passe d'accès
- Attention, le compte peut être verrouillé après plusieurs essais infructueux

Introduction à Linux

Se connecter



Se connecter

- En cliquant sur le nom de l'utilisateur, le mot de passe sera demandé
- Pour la déconnexion, plusieurs possibilités :
 - Ctrl+D ou ^D
 - Logout
 - Exit
 - Se déconnecter en interface graphique

Modifier son mot de passe

- Une fois connecté, il est important de modifier le mot de passe de l'utilisateur
- La commande est la suivante : `$ passwd`
- Cette commande demande l'ancien mot de passe, puis le nouveau deux fois

Introduction à Linux

Se connecter, modifier le mot de passe

Démonstration



Le shell

- Le **shell** est un programme qui **interprète** les **commandes tapées** par l'utilisateur et transmises au système d'exploitation
- Le langage du shell peut être utilisé pour créer des **scripts** grâce à des variables, des boucles, des flux d'entrée-sortie
- Il existe plusieurs shells sur les systèmes Linux

Les différents shells

- Le shell historique est le **Bourne shell** (**sh**), créé par Steve Bourne
- Bill Joy crée le **C shell** (**csh**), basé sur le langage C et intégrant des fonctions comme le rappel de commandes, le contrôle de tâches et les alias
- David Korn reprend le Bourne shell, y inclut les fonctions du C shell et crée le **Korn shell** (**ksh**), shell utilisé sur les systèmes commerciaux Unix

Les différents shells

- Le projet GNU développe une alternative aux shells existants, le **Bourne-Again shell** (**bash**), conforme à la norme POSIX avec quelques extensions
- C'est l'interpréteur de commandes fourni en standard sur les distributions Linux
- C'est ce shell qui nous servira durant tout ce cours

Les variables

- Il existe deux catégories de variables :
 - Les **variables utilisateur** : créées par l'utilisateur
 - Les **variables d'environnement**
- En bash, les variables sont uniquement **de type caractère**
- Pour afficher toutes les variables : **\$ set**
- Pour afficher les variables d'environnement : **\$ env**

Les variables : affectation

- **nomdelavariabile="valeurdelavariabile"**
- Premier caractère alphabétique ou underscore [a-zA-Z_]
- Caractères suivants alphanumériques ou underscore [a-zA-Z0-9_]
- Ne **jamais mettre d'espace** avant ou après le =
- Protéger les espaces dans le contenu des variables (\)
- De préférence, utiliser les doubles quotes pour les chaînes
- Exemple : `nom_cours="Linux et Apache"`

Les variables : exportation

- Une variable n'est pas transmise à un **sous-shell**
- Pour qu'elle soit utilisable il faut l'exporter : `export var`
- Ou exporter et définir en même temps : `export var=valeur`

Les variables : utilisation

- On utilise le symbole `$` devant le nom de la variable
 - `echo $nom_cours`
 - `echo $HOME`
 - `echo $PWD`

Les variables : destruction

- La commande `unset` permet de détruire une variable
- `unset nom_cours`

Les variables d'environnement

- Ce sont des variables définies par le shell
- On trouve par exemple :
 - **HOME** : répertoire de connexion
 - **PWD** : répertoire courant
 - **LOGNAME** : nom de connexion de l'utilisateur
- Par convention, le nom d'une variable d'environnement est en majuscules

Introduction à Linux

Créer une variable

Démonstration



Help !

- Sous Linux, il existe une grande variété de commandes et ces commandes ont beaucoup d'options, pour certaines
- Il faut donc savoir où trouver de l'aide pour les utiliser
- On peut trouver de l'aide en local sur l'installation de la distribution, les commandes pouvant avoir à cette fin une option `--help` ou `-h` :
`$ password --help`

Man

- L'autre outil indispensable sous Linux est le man
- C'est l'aide en ligne des commandes Linux. Son utilisation est simple :
`$ man commande`
- Certaines pages sont traduites, mais beaucoup d'autres sont en anglais

Introduction à Linux

Man

- Il existe plusieurs sections dans le man :
 - Programmes exécutables ou commandes de l'interpréteur de commandes (shell)
 - Appels système (fonctions fournies par le noyau)
 - Appels de bibliothèque (fonctions fournies par les bibliothèques des programmes)
 - Fichiers spéciaux (situés généralement dans `/dev`)
 - Formats des fichiers et conventions. Par exemple `/etc/passwd`
 - Jeux
 - Divers (y compris les macropaquets et les conventions), par exemple `man(7)`, `groff(7)`
 - Commandes de gestion du système (généralement réservées au super-utilisateur)
 - Sous-programmes du noyau [hors standard]

Man

- Par exemple, pour obtenir des informations sur le fichier `/etc/passwd` :
`$ man -s 5 passwd`
- Pour faire une recherche à partir d'un mot-clé :
`$ man -k password`
- man est basé sur le fonctionnement de **vi**, il est donc possible de faire une recherche dans une page :
`/<chaîne à rechercher>`

Introduction à Linux

Utiliser l'aide sur les commandes

Démonstration



Conclusion

- Vous avez découvert le système d'exploitation GNU/Linux
- Vous avez appris à utiliser le shell et découvert les commande de base