

Etudier et s'autoévaluer en vue d'obtenir le "Linux Essentials Certificate of Achievement"

Document en construction ! (document de travail à usage interne : [lpi_linux_essential](#))

Le document proposé sur ces pages est également disponible sur le [wiki de l'ASBL LoLiGrUB](#), et bénéficie de l'apport d'autres auteurs membres de l'ASBL (Thierry G.,...)

Définition de la certification

Cursus LPI Linux Essentials Certificate of Achievement

Objectifs

0 Notions de base en informatique

Cf. la page du cours libre [Initiation à l'informatique](#)

1 La communauté Linux et une carrière dans l'Open Source

Pour ce qui concerne les systèmes d'exploitation accessibles au grand public, l'éditeur de logiciel [Microsoft](#) propose son OS propriétaire, [Windows](#), qui domine en terme de parts de marché. A côté, le constructeur d'ordinateurs [Apple](#) propose son OS, [OS X](#), sur ses Mac et ses systèmes nomades.

En dehors de ces systèmes propriétaires, Linux est un OS sous licence libre, absolument indépendant des constructeurs, et fonctionnant sur la plupart des matériels. Une communauté assez complexe allant des développeurs aux utilisateurs s'est créée autour de ce projet libre, gérant un écosystème particulièrement complexe d'applications, des plus simples aux très complexes. Ces projets peuvent être élaborés dans un cadre purement communautaire et bénévole, ou dans des entités fonctionnant sur des principes d'entreprises, incluant des aspects commerciaux, et par conséquent des perspectives de carrières professionnelles.

1.1 Evolution de Linux et des distributions populaires

La naissance de [Linux](#) date de 1991, grâce à l'impulsion de son créateur, [Linus Torvalds](#), qui a été très vite aidé par de nombreux volontaires séduits par le mode de développement communautaire de ce noyau de système d'exploitation respectant les fonctionnalités de [UNIX](#) et par sa licence de distribution libre, la [GNU GPL](#). Pour constituer véritablement un système d'exploitation complet, il fallait adjoindre d'autres éléments à ce noyau, qui était précisément la pièce manquante du projet [GNU](#), porté par [Richard Stallman](#) depuis 1983, qui comportait déjà les autres éléments essentiels : le compilateur [GCC](#), l'éditeur de texte [Emacs](#), un interface en ligne de commande ([shell](#)), des [bibliothèques logicielles](#), les [commandes Unix](#), des outils réseau, [le serveur graphique X](#) ... finalement, en réunissant ces deux projets, le système d'exploitation libre [GNU/Linux](#) était né, et il entamait alors une formidable progression au milieu des années nonantes.

Dès cette époque, Linux est devenu un OS principalement utilisé pour des serveurs. L'utilisation sur le poste utilisateur était limitée à une catégorie d'utilisateurs férus d'informatique, de geeks.

Plus tard, le développement d'environnements de bureau suffisamment matures a simplifié l'accès à Linux pour de nombreux "profanes", l'administration et les réglages du système devenant accessibles par des [interfaces utilisateurs graphiques](#). Dès les années 2005, il devenait clair que Linux pouvait satisfaire les besoins d'un [utilisateur lambda](#). Nombre d'entre eux n'ont pas hésité à faire le pas, déçus par le système d'exploitation qui s'était imposé lors de l'achat de leur ordinateur.

A l'aube des années 2010 Linux a encore franchi une étape supplémentaire dans la popularité, en prenant au travers du système Android une place de leader sur le marché des smartphones et des tablettes. Connaissant aussi ses nombreuses utilisations dans des appareillages spécifiques : routeurs, décodeurs, télévisions, GPS, mediacenter, on peut à présent affirmer que Linux est présent de manière significative dans chaque foyer.

Pour celui qui voudra installer, utiliser et comprendre un système Linux complet, le choix d'une [distribution](#) est un passage obligé. Les distributions proposent non seulement un environnement complet, mais aussi les moyens de l'étendre par l'ajout de programmes, ou de la mettre à jour. Des communautés d'utilisateurs et de développeurs lui sont souvent associées, ainsi que des sociétés prestataires de services commerciaux.

Toutes semblables, toutes différentes, elles se distinguent entre elles par divers critères, dont ceux-ci :

- l'environnement de bureau par défaut (GNOME, KDE, LXDE, XFCE,...)
- un [système de gestion de paquets de logiciels](#)
- le nombre de paquets proposés
- Un modèle de développement communautaire, ou reposant sur une société commerciale
- le rythme de sorties des versions
- l'existence de versions plus stables, suivies sur le long terme
- des supports matériels plus ou moins important

Les principales distributions :

- [Red Hat](#)
- [Debian](#)
- [SUSE](#)
- [Ubuntu](#) * [http://fr.wikipedia.org/wiki/Linux_Mint|Linux Mint]

- [ArchLinux](#)
- [Gentoo](#)

Pour un comparatif des distributions, cf. le site [Distrowatch](#).

1.2 Principales applications Open Source

1.3 Comprendre les logiciels Open Source et leurs licences

1.4 Compétences ICT (Information & communications technology) et travail sous Linux

2 Trouver son chemin sur un système Linux (pondération : 8)

2.1 Les bases de la ligne de commande

2.1.1 Introduction

Les utilisateurs de Windows qui passent à Linux risquent d'être déstabilisés à la vue de l'interface «ligne de commandes».

Toutefois les versions récentes de Linux intègrent également une interface graphique équivalente à celle utilisée dans Windows, Mac OSX

L'objectif est de vous familiariser avec l'interface texte (ou terminal) pour utilisateur, connue sous le nom «Shell».

Le « Shell » est un moyen pratique pour réaliser des opérations qui peuvent être difficile à exprimer graphiquement.

2.1.2 Le «Shell», c'est quoi?

Les utilisateurs ne peuvent pas communiquer directement avec le noyau du système d'exploitation. En fait cela n'est seulement possible qu'à travers des programmes, comme le «Shell», qui y accède via les «appels systèmes (system call interface)». Le Shell est donc un programme spécial pour utilisateur. Il va lire les commandes introduites au clavier et les interpréter comme commandes à être exécutées. Le Shell agit donc comme «interface» vis à vis de l'ordinateur qui enferme le système d'exploitation un peu comme une coquille et le cache ainsi de l'utilisateur. Le « Shell » est un des nombreux programmes qui accèdent au système d'exploitation.



La courbe d'apprentissage du Shell est plus longue parce que moins intuitive, mais sur certains points comme le traitement par lots de plusieurs fichiers, la ligne de commande conserve un avantage évident.

Types de Shell :

le /bin/sh [shell Bourne](#)

le /bin/bash [shell Bourne Again Shell](#)

le /bin/csh [C shell](#)

le /bin/ksh [Korn shell](#)

le /bin/tcsh [C shell amélioré](#)

Le principe de base est toujours resté le même : Les shells sont des interpréteurs, ils lisent chaque commande saisie par l'utilisateur, vérifient et traitent la syntaxe pour l'exécuter.

La plupart des Shells peuvent également lire des fichiers contenant des séquences de commandes préparées. De tels fichiers sont appelé des « scripts shell ».

Un Shell utilise les étapes suivantes :

- 1 lire une commande du terminal (ou d'un fichier)
- 2 valider la commande
- 3 lancer directement la commande ou démarrer le programme correspondant
- 4 afficher le résultat à l'écran (ou autre part)
- 5 continuer à l'étape 1.

En plus de cette boucle standard, un shell contient un langage de programmation qui inclut des structures plus complexes comme les boucles, conditions et variables.

Ici, nous utiliserons le Shell Bash, l'un des plus couramment utilisé sur les systèmes [GNU/Linux](#). Bash est un logiciel libre publié sous [GNU GPL](#).

Lancer un Shell (mode console ou terminal)

Plusieurs possibilités sont offertes :

- utiliser le menu du bureau (Gnome, Kde, etc.).
- utiliser le menu lancer une application.
 - Dans la fenêtre ainsi ouverte, taper le nom de terminal et valider.
 - La fenêtre **lancer une application** peut être ouverte avec Alt+F2
- Utiliser les consoles virtuelles (il y en a 6).
 - La console virtuelle (tty1 à tty6) est un écran noir où une invite de commande apparaît, de la forme utilisateur@machine ~ \$

Depuis l'interface graphique, il est possible de se connecter à une console virtuelle en utilisant

la combinaison de touches Ctrl+Alt+FN, où N est un chiffre de 1 à 6.

Pour revenir au mode graphique depuis une console virtuelle, utiliser la combinaison de touches ALT+F7.

EXERCICES

- Quel est l'émulateur de terminal (aussi appelé console virtuelle ou terminal virtuel) de votre distribution ?

2.1.3 Les commandes.

Pourquoi des commandes?

Le fonctionnement d'un ordinateur, indépendamment de son système d'exploitation, peut être décrit en trois étapes :

1. l'ordinateur attend les données de l'utilisateur
2. l'utilisateur sélectionne une commande et l'introduit au clavier ou à la souris
3. l'ordinateur exécute la commande

Dans un système Linux, le Shell affiche un chemin (prompt), indiquant ainsi qu'une commande peut être introduite.

Ce « prompt » contient le nom de l'utilisateur, le nom de l'ordinateur sur lequel l'utilisateur est connecté, le nom du répertoire et un caractère final.

```
utilisateur@machine ~ $
```

- utilisateur → représente l'identifiant ou le nom de l'utilisateur connecté
- machine → représente le nom de la machine sur laquelle l'utilisateur est connecté
- ~ → est un raccourci qui signifie le répertoire personnel /home/utilisateur
- \$ → signifie que vous êtes connecté en tant qu'utilisateur

Structure d'une commande

une commande est essentiellement une suite de mots (inspirés vaguement de l'anglais) qui se terminent par l'appui sur la touche ↵ (enter).

Pour que le Shell puisse interpréter cette suite de mots, ceux-ci doivent suivre une syntaxe. Le premier mot de la ligne est souvent le nom de la commande et les autres mots sont les paramètres qui expliquent ce qui est demandé en détail.

A ce stade il est important de savoir que **le shell fait la distinction entre les majuscules et les minuscules** !

Le séparateur entre le nom de la commande et les paramètres peut être un espace ou une tabulation. Notez que vous pouvez également utiliser ↵ (Enter) comme séparateur, à condition de le faire précéder immédiatement de \ afin que le Shell ne l'interprète pas comme une fin de commande.

Les paramètres d'une commande peuvent être divisés en deux types :

- les options : elles sont précédées d'un tiret « - » et peuvent être considérées comme des « switches » qui autorisent ou pas certaines actions. Si vous souhaitez passer plusieurs options à une commande vous pouvez taper la séquence suivante «-a -l -F» ou «-a-l-F». Les options longues débutent le plus souvent avec deux tirets mais ne peuvent pas être regroupées : « foo

-bar -baz »

- les arguments : les paramètres non précédés de tiret sont appelés les arguments. Ils correspondent souvent au nom de fichiers que la commande doit traiter.

la structure générale d'une commande peut être expliquée comme suit :

- Command—“Que faut-il faire?”
- Options—“Comment le faire?”
- Arguments—“Sur quoi faut-il agir?”

Types de Commandes

Dans les shells il y a essentiellement deux types de commandes :

- **Commandes Internes.** Ces commandes sont disponibles dans le Shell lui-même. Le Bash (Bourne-again shell) contient approximativement 30 commandes de ce type. Les commandes comme « exit » et « cd » modifient l'état du shell et donc ne peuvent venir de l'extérieur.
- **Commandes Externes.** Le Shell n'exécute pas directement ces commandes mais il charge des fichiers exécutables qui, dans la structure du système de fichiers Linux, sont souvent situés dans les répertoires /bin ou /usr/bin

En tant qu'utilisateur, vous pouvez utiliser vos propres programmes que le Shell exécutera en tant que commande externe.

EXERCICES

- Ouvrez un terminal. Sur quelle machine êtes vous connecté ? Quel est le nom du répertoire courant ?
- labomons@labomons-N130:~\$ echo comment ce texte va être affiché ?
- la commande « cal » est une commande interne ou externe ?
- La commande « date » affiche par exemple « dimanche 7 octobre 2012, 17:17:14 (UTC+0200) » dans le terminal. Comment faire pour, au départ de la même commande, n'afficher que le jour complet de la semaine (lundi, mardi,...ou dimanche)?
- utilisez la commande « date » pour affichez dans le terminal le texte suivant : nous sommes un « jour complet semaine » du mois « nom complet du mois ». *Exemple: nous sommes un **dimanche** du mois **octobre**.*

2.2 Utiliser la ligne de commande pour obtenir de l'aide

2.3 Utiliser les répertoires et listings de fichiers

2.4 Créer, déplacer et effacer des fichiers

3 La puissance de la ligne de commande (pondération : 10)

3.1 Sauvegarder des fichiers en ligne de commande

3.2 Recherche et extraction de données de fichiers

3.3 Convertir des commandes en script

4 Le système d'exploitation Linux (pondération : 8)

4.1 Choisir un système d'exploitation

4.2 Comprendre le fonctionnement matériel d'un ordinateur

4.3 Les endroits où sont stockées les données

4.4 Votre ordinateur sur le réseau

5 Sécurité et permissions de fichiers (pondération : 7)

5.1 Bases de la sécurité et identification des types d'utilisateurs

5.2 Créer des utilisateurs et des groupes

5.3 Gérer les permissions et propriétaires des fichiers

5.4 Répertoires et fichiers spéciaux

Références

- <http://www.lpi.org/linux-certifications/introductory-programs/linux-essentials>
- [Annonce](#)
- [Document de préparation](#) (version anglaise en bas de page), non libre !
- [The Linux System Administrator's Guide](#), document sous "GNU Free Documentation License"
- <http://www.linuxquestions.org>
- [Linux par l'exemple : Notions de base et configurations types](#), document sous licence GPL de 1999

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

https://dvillers.umons.ac.be/wiki/floss:lpi_linux_essentials?rev=1350166791

Last update: **2012/10/14 00:19**

