

Chapitre 1 - Rappels sur le système GNU/Linux

1.0

Administration système



GORGOUmack SAMBE

1 Avril 2022

Table des matières

Objectifs	3
Introduction	4
I. Le système GNU/Linux	5
1. Historique et caractéristiques du système GNU/Linux.....	5
2. Le noyau Linux.....	5
3. Les outils GNU.....	7
II. Les interfaces du système GNU/Linux	8
1. La ligne de commande.....	8
2. Le mode graphique.....	9
III. Les commandes UNIX	12
1. Les commandes UNIX.....	12
2. Paramètres de commande.....	12
3. Options de commande.....	13
4. La variable d'environnement PATH.....	13
5. Complétion et historique des commandes.....	14
IV. Les pages de manuel UNIX	16
1. Structure des pages de manuel.....	16
2. La commande man.....	16
3. Disposition d'une page de manuel.....	17
V. Les distributions Linux	18
1. Historique et définition.....	18
2. Les distributions GNU/Linux.....	18
Conclusion	20

Objectifs

A l'issue de ce chapitre, l'apprenant doit être capable de :

1. distinguer les **caractéristiques** et **composants** du système GNU/Linux;
2. distinguer et utiliser les éléments de l'**environnement de bureau**;
3. utiliser la **ligne de commande**;
4. lancer des **commandes UNIX** avec paramètres et options ;
5. utiliser de manière adéquate les **pages de manuel** ;
6. distinguer les **distributions** et leurs spécificités.

Introduction

GNU/Linux est un système d'exploitation libre supporté par une grande communauté d'utilisateurs. Plusieurs distributions du système GNU/Linux (Debian, Ubuntu,...) ayant chacune une cible spécifique avec un système d'installation dédié sont aujourd'hui disponibles sur le marché. Nous allons dans ce chapitre faire une présentation du système GNU/Linux et de la distribution Ubuntu que nous utilisons, nous aborderons les points suivants :

1. le système GNU/Linux : le noyau Linux et les outils GNU ;
2. les interfaces du système : la ligne de commandes et le mode graphique ;
3. les commandes UNIX ;
4. les pages de manuels ;
5. les distributions Linux.

I. Le système GNU/Linux

Dans cette section, nous aborderons d'abord l'historique et les caractéristiques du système GNU/Linux, ensuite le noyau Linux et enfin les outils GNU.

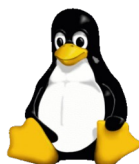
1. Historique et caractéristiques du système GNU/Linux

La naissance du système

En 1991, le projet GNU est riche de très nombreux outils libres mais manque un élément central : le **noyau** du système qui gère le processeur, la mémoire, les périphériques, ...

Linus Torvald, alors jeune étudiant finlandais en Master d'informatique à l'université d'Helsinki, s'inspirant du système *Minix*¹ d'Andrew Tanenbaum, propose pour son mémoire un noyau de système d'exploitation clone d'UNIX qu'il met sous licence GNU GPL : **Linux est né**. Il faut rappeler que Torvald est aussi à l'origine du système de gestion de versions décentralisée *Git*².

GNU/Linux est donc un système d'exploitation libre constitué du noyau Linux et d'un ensemble d'outils GNU.



TUX la mascotte de Linux

Caractéristiques du système GNU/Linux

GNU/Linux est un système d'exploitation **libre** avec une grande communauté et qui respecte la **norme POSIX**³. Il s'appuie sur le **noyau Linux** et un ensemble d'**outils du projet GNU**. Linux désigne seulement le noyau qui est utilisé par d'autres systèmes qui ne s'appuient pas sur GNU tel que le système Android de Google.

GNU/Linux est un système **multitâches** et **multi-utilisateurs**. Il est **multiplateforme**. Il a été initialement conçu pour une architecture de processeur x86 (i386 et x86_64) et a ensuite été porté sur d'autres architectures : POWERPC (Macintosh), AMD64, ARM, ... sans oublier les systèmes embarqués, les supercalculateurs et les téléphones mobiles.

Vous pouvez accéder à une carte interactive du système GNU/Linux *ici*⁴.

2. Le noyau Linux

The Linux Kernel

Linux (The Linux Kernel) désigne le noyau du système GNU/Linux, c'est un *noyau monolithique modulaire*⁵ que vous pouvez télécharger *ici*⁶, son code source est maintenu sur github⁷.

1 - <https://www.minix3.org/>

2 - <https://git-scm.com/>

3 - <https://pubs.opengroup.org/onlinepubs/9699919799.2018edition/>

4 - <http://www.makelinux.net/system/>

5 - https://fr.wikipedia.org/wiki/Noyau_de_syst%C3%A8me_d%27exploitation#Noyaux_monolithiques_non_modulaires

6 - <https://www.kernel.org/>

7 - <https://github.com/torvalds/linux>

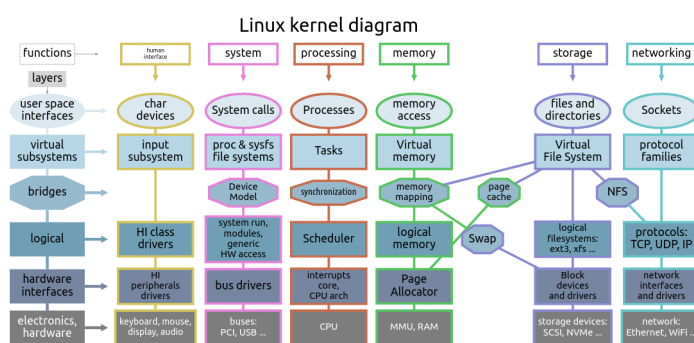
Noyau monolithique

Dans les noyaux monolithiques modulaires, seules les parties fondamentales du système sont regroupées dans un bloc de code unique (monolithique). Les autres fonctions, comme les pilotes matériels, sont regroupées en différents modules qui peuvent être séparés tant du point de vue du code que du point de vue binaire.



Noyau monolithique modulaire

Diagramme du noyau



Architecture du noyau Linux

Le noyau est ainsi développé en couche et de manière modulaire. Vous avez un aperçu des différentes couches du noyau *ici*⁸.

Vous pouvez accéder à une carte interactive du noyau *ici*⁹.

Modules du noyau

Citons **quelques modules** du noyau :

1. Le **module de gestion des fichiers** qui permet de gérer les **systèmes de stockage** (disque dur, clé usb), il est composé de **deux couches** : la couche de **système de fichiers virtuel** et celle des **systèmes de fichiers physiques**.
2. le **module de gestion des processus** qui gère le processeur. L'**ordonnanceur de processus (scheduler)** est l'un des éléments les plus importants de ce module.
3. le module de gestion de la **mémoire virtuelle** qui gère la mémoire vive (RAM) et les mémoires virtuelles (Swap).
4. ...

8 - <https://makelinux.github.io/kernel/diagram/>

9 - <https://makelinux.github.io/kernel/map/>

3. Les outils GNU

Un **logiciel GNU** (paquet GNU, programme GNU) est un logiciel développé sous les auspices du **projet GNU**, il est **toujours libre** avec des droits détenus par la Free Software Foundation ou par l'auteur (personne ou structure). Une bonne partie des logiciels GNU sont soumis à un **copyleft**. Vous trouverez la liste des logiciels GNU *ici*¹⁰. Les manuels en ligne des outils GNU sont disponibles *ici*¹¹.

10 - <https://www.gnu.org/manual/blurbs.html>

11 - <https://www.gnu.org/manual/manual.html>

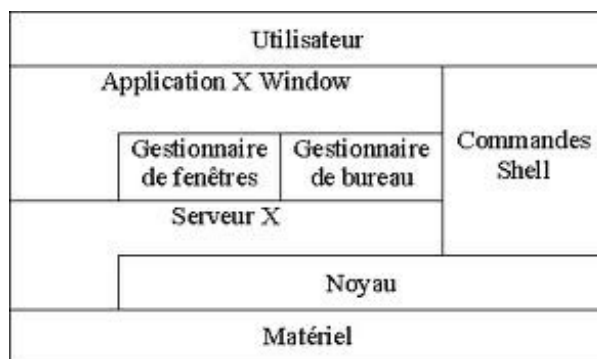
II. Les interfaces du système GNU/Linux

Les interfaces

Pour interagir avec le système GNU/Linux, l'utilisateur a **deux moyens** :

1. la **ligne de commande** à travers une **console** ou un **terminal**. les commandes saisies par l'utilisateur seront interprétées par un programme appelé **shell** ou **interpréteur de commande**.
2. le **mode graphique** à travers un **environnement de bureau**.

Ces deux modes sont complémentaires mais pour un informaticien la maîtrise de la ligne de commande est importante. Dans ce cours, nous travaillerons essentiellement en ligne de commande. Le programme analogue à la ligne de commande Linux dans un environnement Windows est cmd.



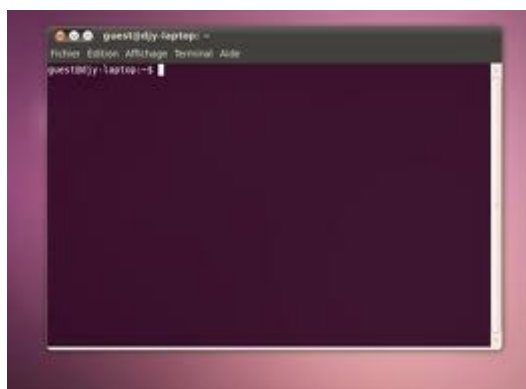
Interaction avec le système

1. La ligne de commande

Le shell

La **ligne de commande** est le moyen le plus simple pour interagir avec le système. La soumission de commandes au système se fait à travers une **console** qui est l'interface textuelle du système ou à travers un **terminal** (émulation d'une console en mode graphique). La console ou le terminal affichent une **invite de commande (prompt)** qui se termine par un **\$** pour un utilisateur standard ou un **#** pour l'administrateur du système (root). Le prompt indique à l'utilisateur que le système attend une commande.

Les commandes tapées au clavier sont interprétées/exécutées par un programme dénommé **interpréteur de commande** ou encore **Shell**, il existe plusieurs shell sur un système GNU/Linux. Les shell permettent également d'automatiser des tâches à travers des scripts, ils possèdent un langage.



Terminal (<https://doc.ubuntu-fr.org/terminal>).

Exemple : Exemples de shell

Plusieurs shell sont disponibles sur les système GNU/Linux, l'utilisateur est libre d'utiliser le shell qu'il souhaite :

1. **sh** (bourne shell) apparu en 1977 et développé par Stephen Bourne est le shell par défaut des systèmes Unix. Il est aussi le shell par défaut de l'administrateur dans la plupart des GNU/Linux.
2. **csh** (C shell) et sa version améliorée **tcsh** sont des extensions de sh qui utilisent une syntaxe plus proche du Langage C
3. **ksh**(Korn shell) apparu en 1983 et développé par David Korn est aussi une extension de sh qui supporte la programmation objet et dans certaines versions l'arithmétique à virgule flottante.
4. **bash**¹² (bourne again shell) est un mixte d'amélioration de sh, ksh et csh. C'est le shell du projet GNU, sa première implémentation a été publiée en 1988. C'est le shell par défaut sur la plupart des systèmes GNU/Linux.
5. ...

Rappelons que les shell, en plus d'interpréter les commandes permettent de déclarer des variables, d'utiliser des structures de contrôle et itératives...ils possèdent un langage.

2. Le mode graphique

Les environnements de bureau

les **environnement de bureau** permettent de travailler en **mode graphique**. Un environnement de bureau est un ensemble de programmes qui permettent de manipuler l'ordinateur à travers une interface graphique qui fait analogie à un bureau. Le **bureau** qui apparaît en arrière-plan de votre écran affiche une image d'arrière-plan et un ensemble d'icônes. Un environnement de bureaux est constitué des éléments suivants :

1. le **système de fenêtrage (display server)** qui fournit des primitives graphiques telles que le rendu de polices de caractères, le tracé de lignes. Linux utilise une implémentation libre du système de fenêtrage X Window (X11 ou X) : *Xorg*¹³.
2. le **gestionnaire de fenêtres (windows manager)** qui est chargé de l'affichage et du placement des fenêtres. Il crée les cadres entourant les fenêtres, zones d'échange entre vous et votre ordinateur, les barres de menus et les panneaux associés qui vous permettent d'accéder à vos logiciels, d'afficher l'heure, de lister vos fenêtres ouvertes, etc. Il gère également les bureaux virtuels et les raccourcis clavier. Compiz, KWin , Metacity sont des exemples de gestionnaire de fenêtres
3. le **gestionnaire d'affichage (display manager)** qui gère vos sessions. KDE Display Manager (KDM) et Gnome Display Manager(GDM) sont des exemples de gestionnaire d'affichage.

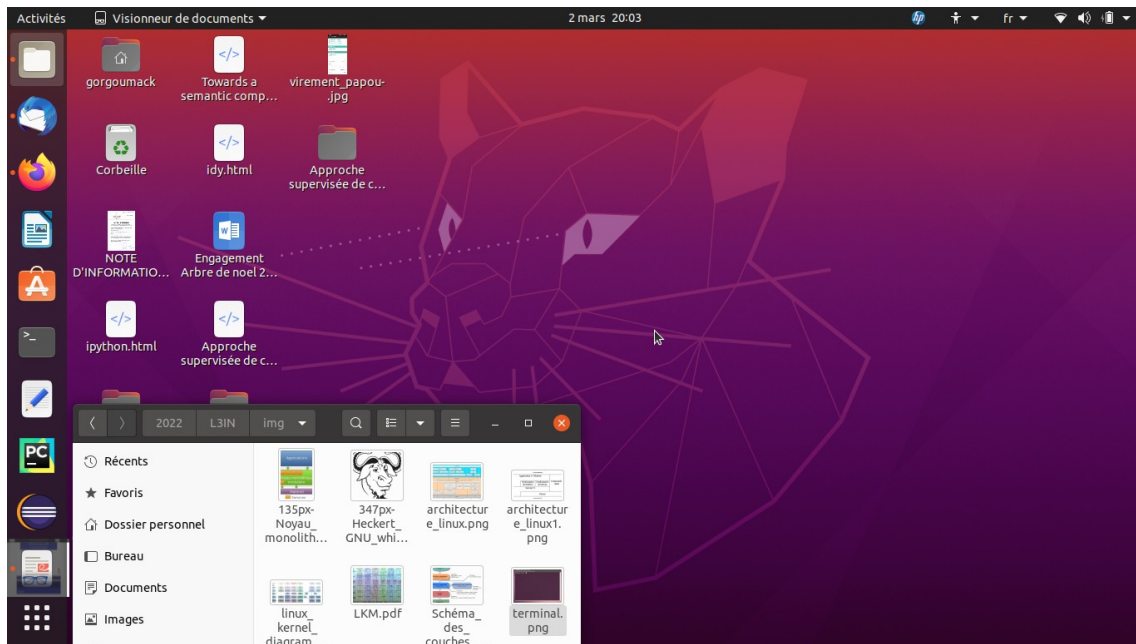
Ils se reposent sur des **bibliothèques de composants graphiques** (Qt¹⁴, GTK¹⁵)

12 - <https://www.gnu.org/software/bash/>

13 - <https://www.x.org>

14 - <https://www.qt.io/>

15 - <https://www.gtk.org/>



Le bureau Gnome

Exemple : Exemples d'environnements de bureau

Sur les système GNU/Linux, les environnements de bureau sont interchangeables, ils ne sont qu'une couche d'interface au dessus du système, l'utilisateur est libre d'utiliser l'environnement de bureau qu'il souhaite.

Parmi les environnements les plus connus nous pouvons citer :

1. *Mate*¹⁶
2. *Gnome*¹⁷
3. *KDE*¹⁸
4. *xfce*¹⁹

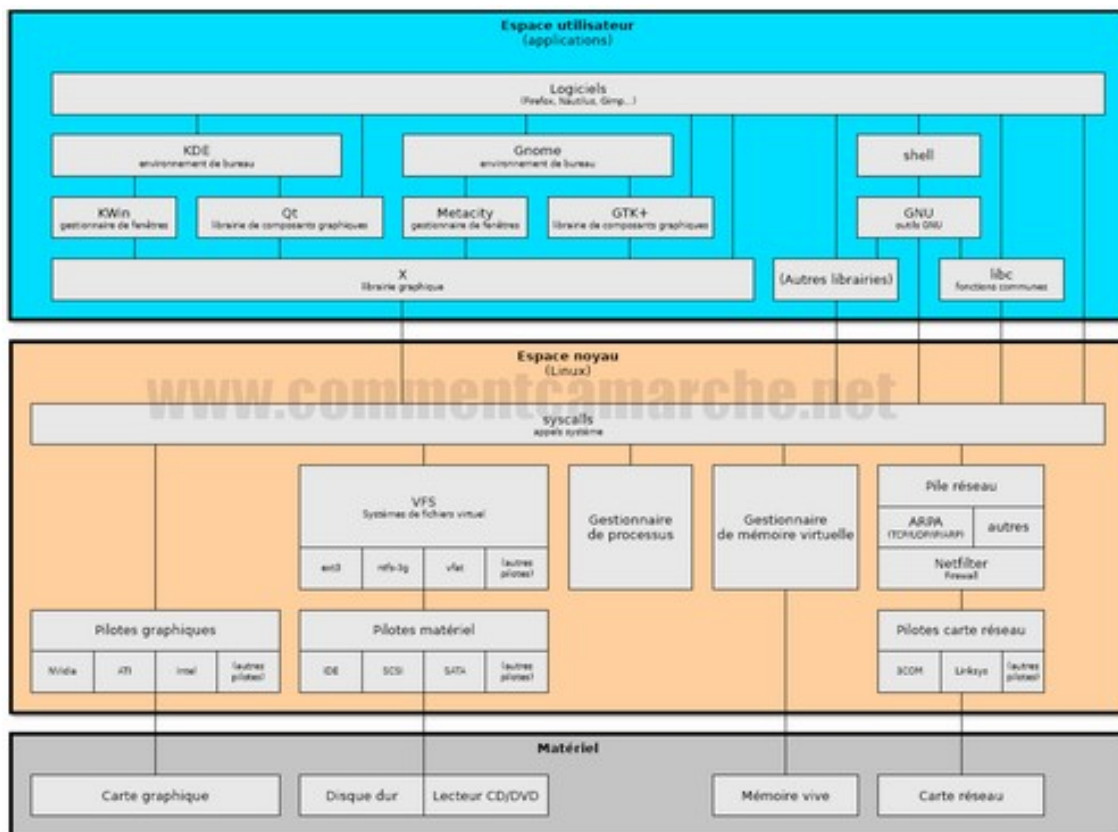
16 - <https://mate-desktop.org/fr/>

17 - <https://www.gnome.org/>

18 - <https://kde.org/fr/>

19 - <https://xfce.org/>

Exemple : Architecture illustrée d'exemples d'outils



Interaction avec le système (Illustré)

III. Les commandes UNIX

Nous aborderons dans cette section les commandes UNIX, les paramètres et les options de commandes, la variable d'environnement PATH et enfin l'historique de commande et les mécanismes de complétion de commande.

1. Les commandes UNIX

Commande

Une **commande** est un fichier **exécutable** que l'on lance en tapant son patronyme sur une **ligne de commande** et qui est destiné à réaliser une **action précise**.

Les systèmes UNIX mettent à disposition des centaines de commandes. Certaines commandes sont intégrées au shell et d'autres sont installées sur le système.

Exemple : Exemples de noms de commandes

Quelques exemples de commande :

1. **lscpu** : afficher des informations sur l'architecture du processeur.
2. **date** : afficher ou configurer la date et l'heure
3. **cal** : affiche le calendrier
4. **echo** : afficher une ligne de texte
5. **mkdir** : créer des répertoires
6. **ls** : afficher le contenu de répertoires

Syntaxe d'une commande UNIX

Une commande soumise au système est constituée d'un **nom** suivi d'une liste (éventuelle) d'**options** et d'une liste (éventuelle) de **paramètres**.

nom [-options] [argument1...]

Certaines commandes ont des paramètres obligatoires, d'autres ont des options obligatoires ou les deux. Certaines commandes ne prennent pas de paramètres, d'autres ne prennent pas d'options.

Exemple : Exemples de commandes

```
$ date #affiche la date et l'heure  
$ ls #liste le contenu du répertoire courant  
$ cal #affiche le calendrier du mois en cours  
$ clear #efface l'écran de terminal
```

Remarque : le commentaire sur le shell bash

le caractère **#** permet de faire un **commentaire** sur le shell bash

2. Paramètres de commande

Paramètres

Les **paramètres** désignent généralement les éléments auxquels s'appliquent la commande. Cela peut être une chaîne de texte, un fichier, un processus, un utilisateur, une commande...

Exemple : Exemples de commandes avec paramètres

```
$ echo bonjour #affiche le texte bonjour  
$ mkdir repertoire1 #crée le répertoire repertoire1
```

```
$ cal 1996 #affiche le calendrier de 1996  
$ ls repertoire1 #liste le contenu de repertoire1
```

3. Options de commande

Options

Les **options** permettent de modifier le fonctionnement d'une commande. Elles se placent à la suite de la commande et commencent généralement par un tiret. La majorité des options, sous Unix, est composée d'un tiret et d'une seule lettre, on peut les qualifier d'options courtes : -l -a -i -x -r. .

Combinaison d'options

Il est possible de **combiner les options** courtes en les écrivant séparément à la suite de la commande chaque option avec son tiret, par exemple -i -l. Il est aussi possible de les combiner en les mettant ensemble derrière un seul tiret, par exemple -il.

Options longues

Certaines options à nom long sont en général précédés de deux tirets :

1. - -version qui donne la version de la commande ;
2. - -help qui affiche une page d'aide de la commande

Exemple : Exemples de commandes avec options

```
$ ls -l #lister le contenu en affichage long  
$ ls -i #lister le contenu en affichant les inodes  
$ ls -il #lister le contenu en affichage long et avec les inodes  
$ ls -i -l  
$ date - -version #affiche les informations de version de la commande date.  
$ man - -help #affiche de l'aide pour la commande man
```

Exemple : Exemples de commandes avec options et paramètres

```
$ echo -n bonjour la classe #affichera la ligne de texte bonjour la classe mais sans le retour à la ligne de fin de texte.  
$ ls -l repertoire1 #lister le contenu du répertoire repertoire1 en affichage long
```

4. La variable d'environnement PATH

Variable d'environnement

Les **variables d'environnements** sont des variables du système accessibles à tous les programmes. elles permettent ainsi de partager des informations de configuration sans avoir à utiliser des fichiers de configurations. Par exemple, la variable d'environnement « **LANG** » détermine la langue d'interfaçage avec l'utilisateur, elle aura pour valeur probablement fr_FR.UTF-8 pour un utilisateur francophone.

Le path

La **variable d'environnement PATH** contient la liste des répertoires dans lesquels le système recherche une commande saisie par l'utilisateur. Beaucoup d'exécutables sont situés dans les répertoires /bin et /usr/bin (binaries).

Par **exemple**, si votre **PATH** vaut **/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games**, le système cherchera les commandes que vous soumettez respectivement dans /usr/local/sbin, puis dans /usr/local/bin, puis dans /usr/sbin, puis dans /usr/bin, puis dans /sbin:/bin, puis /usr/games, puis dans /usr/local/games.

Si le système ne trouve pas la commande, il vous affichera un message d'erreur du genre **commande introuvable**.

Affichage du path

On peut **afficher la valeur d'une variable** en général, d'une variable d'environnement en particulier avec la commande echo :

```
$ echo $LANG #affichage de la langue d'interfaçage de l'utilisateur
```

```
$ echo $PATH #affichage du PATH de l'utilisateur
```

```
$echo $SHELL # affichage du shell de l'utilisateur
```

5. Complétion et historique des commandes

Sur le shell bash, Un **mécanisme de complétion** et un **historique de commandes** sont disponible pour **faciliter l'usage de la ligne de commande**.

Complétion de commandes

Les touches de **tabulation** (tab) et d'**échappement** (ESCAPE ou esc) offrent un mécanisme de complétion des commandes :

- Lorsque l'utilisateur débute la **saisie d'un nom de commande**, un **appui sur la touche de tabulation** ou **deux appuis sur la touche d'échappement** entraîne l'affichage automatique du **reste du nom de la commande**.
- Si **plusieurs solutions** sont possibles (cas de plusieurs commandes avec même début de nom), le shell **affiche tous les caractères communs** et attend que l'utilisateur continue la frappe.
- Si l'utilisateur tape **une seconde fois sur la tabulation**, le shell affiche la **liste des entrées commençant par le préfixe commun**.
- Enfin, dans le cas où **aucun nom de commande** ne commence par le texte déjà tapé, **un bip est émis**.

Ce mécanisme de complétion s'applique également aux **noms de fichiers** selon le même principe.

Historique des commandes

L'historique autant que la complétion fait gagner beaucoup de temps en ligne de commande. Les **commandes saisies** par l'utilisateur sont conservées dans un **fichier d'historique** (**.bash_history**) et **numérotées**.

La commande **history** sans paramètres affiche **tout l'historique**, elle vide l'historique avec l'option -c. Les **touches de direction** du clavier **haut** (\$\uparrow\$) et **bas** (\$\downarrow\$) permettent respectivement de **remonter** et **descendre** dans l'historique des commandes.

On peut **faire référence à une entrée** de ligne de commande dans la liste de l'historique :

- \$ **!n** : fait référence à la **commande numéro n** ;
- \$ **!-n** : fait référence à la commande en **position n en partant du bas de l'historique**.
- \$ **!!** : fait référence à la **dernière commande exécutée** (équivalent à **!-1**);
- \$ **!chaîne** : fait référence à la **dernière commande** de l'historique **débutant avec la chaîne de caractère chaîne**.

La combinaison de touche **[CTRL]+[r]** fait entrer en **mode recherche** sur l'historique. L'historique permet de **composer une commande** à partir des **entrées de l'historique**. Reportez vous à la page de manuel de history pour plus d'options.

IV. Les pages de manuel UNIX

1. Structure des pages de manuel

Les **pages de manuels (manpages)** sont fondamentales pour tout utilisateur UNIX, c'est la première aide lorsqu'on travaille en ligne de commandes et les pages sont très précises et fournissent toutes les informations dont on peut avoir besoin. elles sont disponibles en **ligne** et sur toute **installation du système**. Elles sont **nativement en anglais** mais on peut les faire passer dans une **autre langue** tel que le français (toutes les pages ne sont pas traduites, c'est au **bénévolat**).

Elles sont accessible en ligne *ici* ²⁰ et en français *ici*²¹.

Les sections des pages de manuel

Les pages de manuel sont organisées en **sections**. Par exemple sur **Linux** nous avons les **9 sections** suivantes (le texte suivant a été emprunté aux manpage de la commande man) :

1. Programmes exécutables ou commandes de l'interpréteur de commandes (shell) ;
2. Appels système (fonctions fournies par le noyau) ;
3. Appels de bibliothèque (fonctions fournies par les bibliothèques des programmes) ;
4. Fichiers spéciaux (situés généralement dans /dev) ;
5. Formats de fichiers et conventions ;
6. Jeux ;
7. Divers (y compris les macropaquets et les conventions), par exemple man(7) ;
8. Commandes de gestion du système (généralement réservées au superutilisateur) ;
9. Sous-programmes du noyau [hors standard].

Le même nom d'une page de manuel peut apparaître dans plusieurs sections, c'est le cas de la commande man qui apparaît en section 1 et en section 7.

Les fichiers

Les pages de manuel se trouvent habituellement dans le répertoire **/usr/share/man** pour ubuntu. Elles sont organisées en une hiérarchie, à raison d'**un répertoire par section du manuel**. Par exemple, les pages de la section 1 sont dans le répertoire man1 et la page de la commande lscpu(1) est dans le fichier man1/lscpu.1. Il est possible que ces pages soient compressées. On peut ainsi trouver un fichier man1/lscpu.1.gz (compression par gzip(1)) ou un fichier man1/lscpu.1.bz2 (compression par bzip2(1)).

2. La commande man

la commande **man** recherche la **page de manuel** associée à son **paramètre** et l'affiche. Il cherche par défaut dans **toutes les sections** disponibles et affiche la **première page trouvée** en suivant un **ordre prédéfini** configurable(par défaut 1 8 3 2 5 4 9 6 7). Le paramètre peut être une commande, un utilitaire, un format, ...

Elle recherche **uniquement dans une section** si c'est précisé en **option**.

Pour afficher la page de manuel d'une commande nomcommande, il suffit d'utiliser la commande suivante :

```
$ man nomcommande
```

Par exemple pour afficher l'aide sur la commande lscpu, il suffit de taper :

```
$ man lscpu
```

On défile dans la page de manuel à l'aide des touches de directions et pour quitter la page de manuel taper la lettre q.

20 - <https://www.kernel.org/doc/man-pages/>

21 - <http://manpagesfr.free.fr/>

Pour afficher la page de manuel d'une commande relativement à une section, il suffit d'utiliser la commande suivante :

\$man section nomcommande

section désigne le numéro de section.

Par exemple

\$man 7 man

affiche la page de manuel de man qui est dans la septième section, section des macropaquets.

\$ man -k mot-clé

Recherche la description courte et le nom des pages de manuel comportant mot-clé (équivalent à \$ apropos mot-clé).

\$ man -k echo

Recherche la description courte et le nom des pages de manuel comportant le mot-clé echo.

\$ man -f commande

affiche la description courte de la page de manuel de commande (équivalent à whatis).

\$ man -f echo

affiche la description courte de la page de manuel de echo.

3. Disposition d'une page de manuel

Toutes les pages de manuels suivent une disposition commune structurée sous la forme de chapitres dont on peut citer les principaux :

- **NAME** : nom de la commande et fonction.
- **SYNOPSIS** : différentes syntaxes de la commande.
- **DESCRIPTION** : Description générale.
- **COPYRIGHT** : droits d'auteurs
- **EXAMPLES** : Exemples d'usage de la commande et suggestions.
- **SEE ALSO** : Références croisées et citations.



```

LSCPU(1)                                Commandes utilisateur                                LSCPU(1)
NOM
    lscpu - Afficher des informations sur l'architecture du processeur
SYNOPSIS
    lscpu [-a|-b|-c] [-x] [-s répertoire] [-e[=liste]] [-p[=liste]]
    lscpu -h|-V
DESCRIPTION
    lscpu collecte des renseignements sur l'architecture processeur à partir de sysfs
    et /proc/cpuinfo. La sortie de la commande peut être optimisée pour l'analyse ou
    pour faciliter la lecture. Par exemple, le nombre de processeurs, de processus
    légers, de cœurs, de sockets et de nœuds NUMA font partie des renseignements. Des
    renseignements sont aussi fournis sur les caches et les partages de cache, la fa-
    mille, le modèle, le BogoMips, le boutisme et la révision.

    Les options ayant pour résultat un tableau en sortie ont un argument liste. Uti-
    lisez cet argument pour personnaliser la sortie de la commande. Indiquez une
    liste d'étiquettes de colonne séparées par des virgules pour limiter le tableau
    en sortie à ces colonnes dans l'ordre indiqué. Consultez COLONNES pour une liste
-- MOST: *stdin*                                (1,1) 0%
Press 'Q' to quit, 'H' for help, and SPACE to scroll.

```

page de manuel de lscpu

V. Les distributions Linux

Nous reviendrons dans cette section sur l'historique des distributions GNU/Linux et leur définition. Nous présenterons ensuite quelques distributions.

1. Historique et définition

Historique des distributions

Le système Linux a lui tout seul est un noyau stable mais vous n'aurez pas dessus une belle interface graphique ni les utilitaires et logiciels dont vous aurez besoin (calculatrice, suite bureautique, lecteur PDF, lecteur multimedia, ...) selon votre usage du système. Les premiers utilisateurs du système GNU/Linux devaient eux même **composer** et **installer** leur système : installer le noyau, installer les outils GNU et outils tiers selon leur usage du système.

Concernant la **composition** de logiciels, il apparaît que les besoins étaient les mêmes selon les types/communautés d'utilisateurs :

- les utilisateurs lambda avaient besoin d'un système convivial, ergonomique avec dessus les outils qu'ils ont l'habitude d'avoir sur d'autres systèmes : suite bureautique, lecteur multimedia, lecteur video, chat, VOIP, ...
- le monde de l'éducation, en plus de ces outils, a besoin d'outils relatifs à la pédagogie et à l'éducation
- un "joueur professionnel" a des besoins différents liés à des aspects de performance, un infographiste également avait plus besoin de l'aspect ergonomie et graphiques et des outils liés à ce domaine.
- sur un serveur, les besoins en termes de services sont beaucoup plus à prendre en compte (serveur dhcp, serveur de fichier, serveur de mail, serveur de fichier, serveur ssh...)
- ...
- les informaticiens avaient besoin d'outils liés à la conception et au développement de logiciels : compilateurs, débogueurs, ...

Concernant l'**installation**, l'installation séparée du noyau, des outils GNU, des utilitaires de bases et des logiciels spécifiques dont l'utilisateur a besoin est une tâche ardue.

Il était nécessaire de mettre en place un système d'installation qui facilite l'installation du noyau, des outils GNU et des logiciels spécifiques selon les usages de l'utilisateur. C'est la naissance des premières distributions

Qu'est ce qu'une distribution Linux

Une **distribution GNU/Linux** désigne un ensemble de logiciels cohérents assemblés autour du noyau, la plupart des distributions possèdent aujourd'hui un système d'installation. Il existe une très grande variété de distribution, chacune ayant un objectif et une philosophie particulière, ils ont en commun le noyau Linux et certaines commandes.

Une distribution peut être vue comme une composition de logiciels autour du noyau Linux accompagné d'un système d'installation.

2. Les distributions GNU/Linux

Il existe aujourd'hui une multitude de distributions GNU/Linux. Cette diversité peut être vue comme un inconvénient mais il laisse à l'utilisateur la possibilité de choisir une distribution selon son usage du système (serveur, client, usage pédagogique, développement logiciel). Un des avantages des distributions est la possibilité que l'utilisateur a d'utiliser dans une même famille de logiciels, des outils différents et/ou des versions différentes : même pour le noyau,

l'utilisateur peut installer plusieurs versions du noyau et démarrer son système avec la version souhaitée.

Nous nous limiterons ici à citer quelques distributions :

- **Debian**²²: Debian est une version entièrement libre et non commercial distribuée par une association : la SPI (Software in the Public Interest). Elle est considérée comme l'une des distributions les plus stables aujourd'hui et comme la distribution des informaticiens/développeurs.
- **Ubuntu**²³: Ubuntu est distribué par la société Canonical, c'est l'une des distributions les plus connus vu la facilité qu'elle offre à l'utilisateur, particulièrement les utilisateurs basculant de Windows. C'est une distribution commerciale basée sur Debian, Canonical fait du profit à travers la vente de services même si elle distribue une version gratuite. elle a beaucoup de distribution dérivées (*Kubuntu*²⁴ pour les adeptes de l'environnement de bureau KDE, ubuntu activant par défaut l'environnement gnome, *Lubuntu*²⁵ version légère qui utilise l'environnement de bureau LXQt²⁶, ...)
- *fedora*²⁷, *opensuse*²⁸, *Elementary OS*²⁹, *Linux Mint*³⁰, *CentOS*³¹,...

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

23 - <https://ubuntu.com>

24 - <https://kubuntu.org/>

25 - <https://lubuntu.net/>

26 - <https://lxqt-project.org/>

27 - <https://getfedora.org/fr/>

28 - <https://fr.opensuse.org>

29 - <https://elementary.io/fr/>

30 - <https://linuxmint.com/>

31 - <https://fr.centos.org>

Conclusion

Dans ce chapitre, nous avons fait un petit tour d'horizon du système GNU/Linux. GNU/Linux est un système d'exploitation libre en constante évolution depuis 1991. La découverte des systèmes d'exploitation par un système libre est très pédagogique, il permet de voir en détail les architectures et choix techniques faits pour la mise en place du système.

Le code source est téléchargeable, il est intéressant d'en lire des parties, il atteint aujourd'hui des millions de lignes. La documentation est très riche à travers les manpages mais aussi les sites web de FSF , de GNU, des différentes distributions et les forums. Le code source du noyau et des outils GNU sont accessibles. Pour toute question sur GNU/Linux et les outils open source, vous trouvez assez facilement une réponse sur un forum de discussions.