

Cours d'introduction a Linux

Introduction

Linux est un système d'exploitation open-source basé sur le noyau Linux. Il est largement utilisé dans le monde entier, tant par les développeurs que par les entreprises, en raison de sa stabilité, de sa sécurité et de sa flexibilité. Ce cours d'introduction à Linux vous donnera un aperçu des concepts de base et des compétences nécessaires pour commencer à utiliser Linux.

Sommaire

Table des matières

Cours d'introduction a Linux	1
Introduction.....	1
Sommaire	2
Historique de Linux.....	3
Origines de Linux	3
Le Contexte des Années 1980	3
Linus Torvalds et le Projet Linux	3
La Collaboration de la Communauté Open Source	4
Évolution de Linux	4
Architecture de Linux	5
Le Noyau Linux.....	5
L'Espace Utilisateur (Userland).....	5
La Communication entre le Noyau et l'Espace Utilisateur	6
Personnalisation et Distributions Linux	6
Distribution Linux	7
Qu'est-ce qu'une distribution Linux ?.....	7
Exemples de distributions populaires (Ubuntu, CentOS, Debian, Fedora).....	8
La ligne de commande.....	14
Gestion des utilisateurs et des permissions	16
Gestion des fichiers et des répertoires.....	18
Éditeurs de texte.....	20
Gestion des packages et des logiciels.....	24
Gestionnaires de paquets (apt, yum)	24
Installation et mise à jour de logiciels	25
Gestion des processus	26
Automatisation des tâches avec les scripts Shell	28

Historique de Linux

Origines de Linux

Linux, un système d'exploitation révolutionnaire et open-source, trouve ses origines dans le monde de l'informatique et de la programmation des années 1980. Pour comprendre comment Linux est né, il est essentiel de connaître l'histoire de son créateur, Linus Torvalds, et le contexte dans lequel il a développé ce système d'exploitation.

Le Contexte des Années 1980

Les années 1980 ont été une période de croissance rapide dans le domaine de l'informatique personnelle. Les ordinateurs personnels (PC) étaient de plus en plus accessibles, mais les systèmes d'exploitation propriétaires, tels que MS-DOS de Microsoft, étaient les plus couramment utilisés. Ces systèmes étaient coûteux et restrictifs, ne permettant pas aux utilisateurs de modifier leur code source. Cela a suscité le désir de développer un système d'exploitation libre et ouvert, où les utilisateurs auraient un contrôle total.

Linus Torvalds et le Projet Linux

Linus Torvalds, un étudiant en informatique finlandais, est né le 28 décembre 1969. En 1991, alors qu'il poursuivait ses études à l'Université de Helsinki, il commença à travailler sur son projet personnel de développement d'un système d'exploitation. Linus était fasciné par le système d'exploitation Unix, mais il ne pouvait pas se permettre les coûts associés à son utilisation.

Le 25 août 1991, Linus Torvalds publia un message sur Usenet, une plateforme de discussion en ligne, annonçant son projet de créer un système d'exploitation semblable à Unix. Il écrivit : *"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)."*

Ce message a marqué le début de ce qui allait devenir le projet Linux. Linus Torvalds a développé un noyau de système d'exploitation, le cœur du système, qui a été appelé le "noyau Linux". Le noyau était basé sur les principes de l'Unix et a rapidement attiré l'attention des développeurs du monde entier.

La Collaboration de la Communauté Open Source

La principale innovation de Linux résidait dans sa nature open source. Linus a autorisé quiconque à contribuer à son projet, à améliorer le noyau et à distribuer des versions modifiées. Cela a attiré l'attention de nombreux programmeurs talentueux qui ont commencé à collaborer sur le projet.

Le noyau Linux a été publié sous la licence GNU General Public License (GPL), qui garantit que toutes les versions modifiées restent open source. Cette licence a été cruciale pour la croissance du projet, car elle a permis à la communauté open source de contribuer sans crainte de voir leur travail approprié par des entités commerciales.

Évolution de Linux

Au fil des ans, Linux est devenu de plus en plus sophistiqué. Il a évolué pour prendre en charge un large éventail d'architectures matérielles, devenant ainsi l'un des systèmes d'exploitation les plus polyvalents au monde. De nombreuses distributions Linux, telles qu'Ubuntu, CentOS, Debian et Fedora, ont vu le jour pour répondre aux besoins variés des utilisateurs.

Linux est désormais omniprésent, utilisé non seulement sur les serveurs et les supercalculateurs, mais aussi sur les ordinateurs personnels, les smartphones (Android), les routeurs, les systèmes embarqués et bien plus encore.

Architecture de Linux

L'architecture de Linux est fondamentale pour comprendre comment ce système d'exploitation fonctionne. Linux repose sur un modèle de système d'exploitation Unix, et il est conçu de manière à être modulaire, flexible, et hautement personnalisable. Cette page vous guidera à travers les principaux composants de l'architecture Linux et comment ils interagissent pour fournir un environnement informatique robuste.

Le Noyau Linux

Le noyau Linux est le cœur du système d'exploitation. Il est responsable de la gestion des ressources matérielles, de la planification des tâches, de la gestion de la mémoire, des opérations d'entrée/sortie, de la sécurité, et bien plus encore. Le noyau est responsable de l'interaction directe avec le matériel de l'ordinateur, ce qui permet aux logiciels et aux utilisateurs d'accéder aux ressources matérielles de manière transparente.

Le noyau Linux est monolithique, ce qui signifie qu'il intègre de nombreuses fonctionnalités essentielles directement dans son propre espace d'exécution. Cependant, il prend également en charge des modules qui peuvent être chargés et déchargés dynamiquement pour étendre ses fonctionnalités sans nécessiter une réinitialisation du système.

L'Espace Utilisateur (Userland)

L'espace utilisateur, comme son nom l'indique, est l'espace où les programmes et les utilisateurs interagissent avec le système. Il comprend divers éléments :

Les Bibliothèques Système

Les bibliothèques système sont des collections de fonctions qui offrent un ensemble de services standard aux programmes. Elles simplifient le développement de logiciels en fournissant un accès commun à des fonctionnalités telles que la gestion des fichiers, la communication réseau, et la gestion de la mémoire.

Les Programmes Utilitaires

Linux propose une multitude de programmes utilitaires en ligne de commande et des interfaces graphiques pour effectuer une variété de tâches. Ces programmes comprennent des éditeurs de texte, des navigateurs web, des outils de gestion des fichiers, et bien d'autres.

Les Interfaces Utilisateur Graphiques (GUI)

Les environnements de bureau, tels que GNOME, KDE, et Xfce, fournissent des interfaces graphiques conviviales pour les utilisateurs finaux. Ils permettent de gérer des fichiers, de lancer des applications, et de personnaliser l'apparence du système.

Les Applications

Les logiciels applicatifs, tels que des navigateurs web, des suites bureautiques, des clients de messagerie, des jeux, etc., sont exécutés dans l'espace utilisateur. Ils sont généralement conçus pour répondre aux besoins spécifiques des utilisateurs.

La Communication entre le Noyau et l'Espace Utilisateur

Linux maintient une interface de programmation (API) bien définie entre le noyau et l'espace utilisateur. Cette API permet aux programmes de demander des services au noyau, tels que la création de fichiers, la gestion des processus, ou la configuration réseau. Elle assure également l'isolation des processus, ce qui signifie que les applications s'exécutent de manière indépendante les unes des autres.

La communication entre le noyau et l'espace utilisateur se fait par le biais de "appels système" (system calls) : des fonctions spéciales qui permettent aux programmes de solliciter le noyau pour effectuer des opérations critiques. Les appels système sont une interface essentielle pour la gestion des ressources et des interactions avec le matériel.

Personnalisation et Distributions Linux

Une caractéristique essentielle de Linux est sa capacité à être hautement personnalisable. Les utilisateurs peuvent créer des configurations sur mesure en choisissant différents composants logiciels, en modifiant les paramètres système, et en installant ou supprimant des modules du noyau en fonction de leurs besoins spécifiques. Cela a conduit à la création de nombreuses distributions Linux, chacune étant adaptée à des cas d'utilisation particuliers.

Distribution Linux

Qu'est-ce qu'une distribution Linux ?

Une distribution Linux, souvent appelée simplement "distro", est une version spécifique du système d'exploitation Linux qui inclut le noyau Linux, des bibliothèques système, des utilitaires, des gestionnaires de paquets, des configurations système, et parfois même un environnement de bureau ou une interface utilisateur. Les distributions Linux sont créées et maintenues par différentes organisations ou communautés de développeurs, chacune ayant ses propres objectifs et philosophies. Chaque distribution est conçue pour répondre à des besoins spécifiques, que ce soit pour les serveurs, les ordinateurs de bureau, les systèmes embarqués, ou d'autres cas d'utilisation. Les distributions Linux offrent également des mécanismes de gestion des logiciels, ce qui permet aux utilisateurs d'installer, mettre à jour et gérer facilement des applications et des packages logiciels. En résumé, une distribution Linux constitue un ensemble complet et prêt à l'emploi du système d'exploitation Linux, adapté à des besoins variés, et souvent accompagné d'un écosystème logiciel cohérent.

Exemples de distributions populaires (Ubuntu, CentOS, Debian, Fedora)

Linux Mint

Linux Mint est une distribution Linux populaire qui est réputée pour sa convivialité et sa facilité d'utilisation. Basée sur Ubuntu, elle propose un environnement de bureau Cinnamon par défaut, qui offre une expérience utilisateur familière et intuitive, similaire à celle de Windows. Linux Mint se distingue également par sa stabilité et sa fiabilité, ce qui en fait un choix apprécié tant pour les débutants que pour les utilisateurs plus avancés.



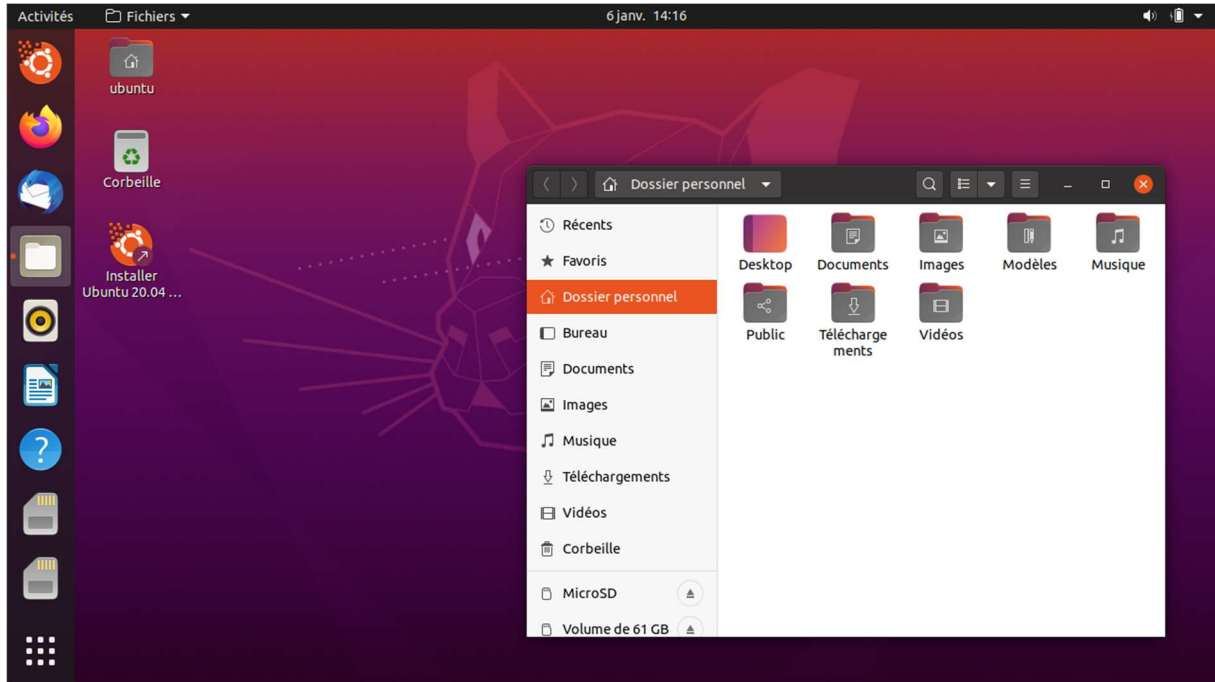
Debian

Debian est l'une des distributions Linux les plus anciennes et les plus respectées. Connu pour son engagement envers la philosophie du logiciel libre, Debian est construit autour de principes de stabilité, de sécurité et de convivialité. Elle se caractérise par son gestionnaire de paquets APT (Advanced Package Tool), qui simplifie la gestion des logiciels et des mises à jour. Debian est également la base de nombreuses autres distributions populaires, dont Ubuntu.



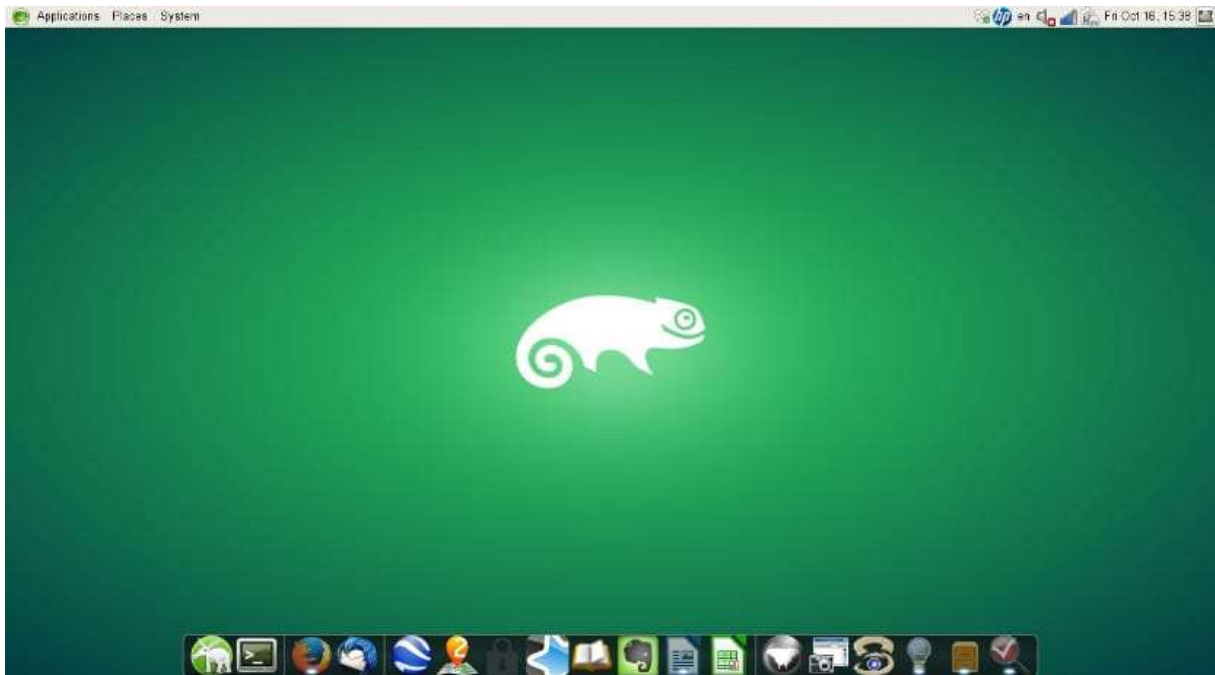
Ubuntu

Ubuntu est une distribution Linux populaire et conviviale basée sur Debian. Une des particularités d'Ubuntu est son modèle de versionnement basé sur le temps, avec des versions régulières publiées tous les six mois et des versions LTS (Long Term Support) tous les deux ans, bénéficiant d'une maintenance et d'un support étendus. Ubuntu comprend par défaut des logiciels non libre, ce qui permet un meilleur support du matériel (nvidia, cartes wifi, ...)



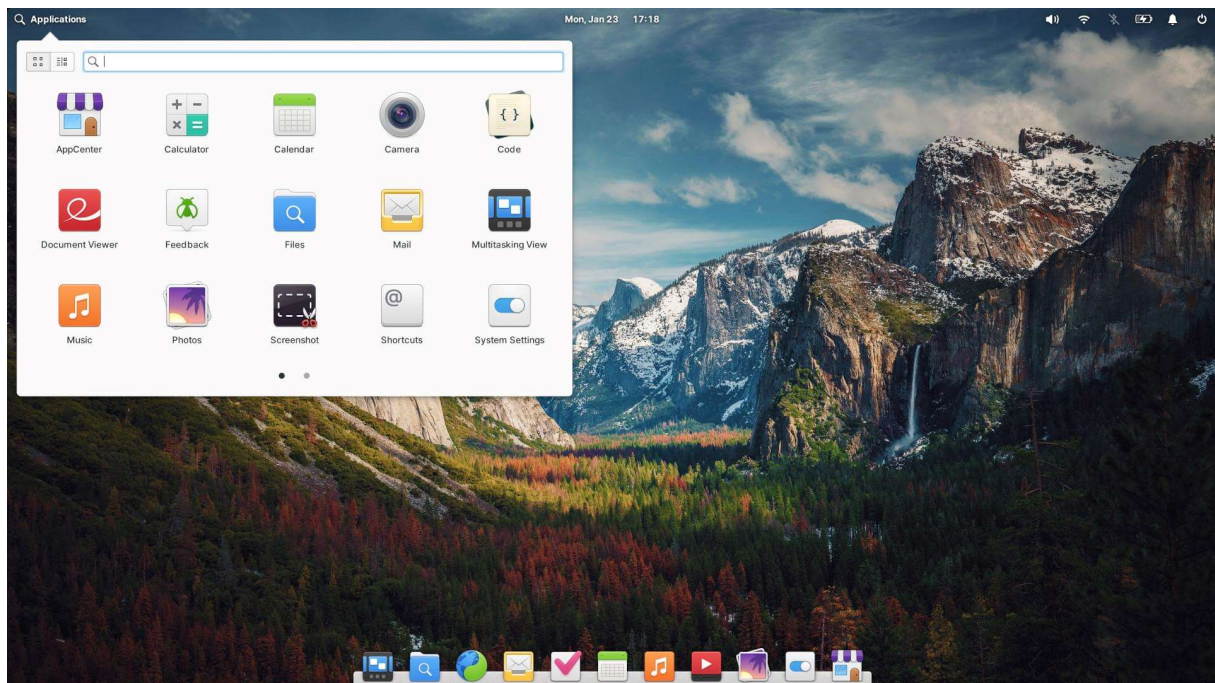
openSUSE / SUSE Linux Enterprise Server

SUSE est une distribution Linux qui se distingue par son engagement envers les solutions d'entreprise. Elle est développée et maintenue par la société SUSE, et dispose d'un support professionnel. SUSE propose des versions spécifiques pour les serveurs, les postes de travail et les systèmes embarqués, répondant ainsi aux besoins variés des entreprises.



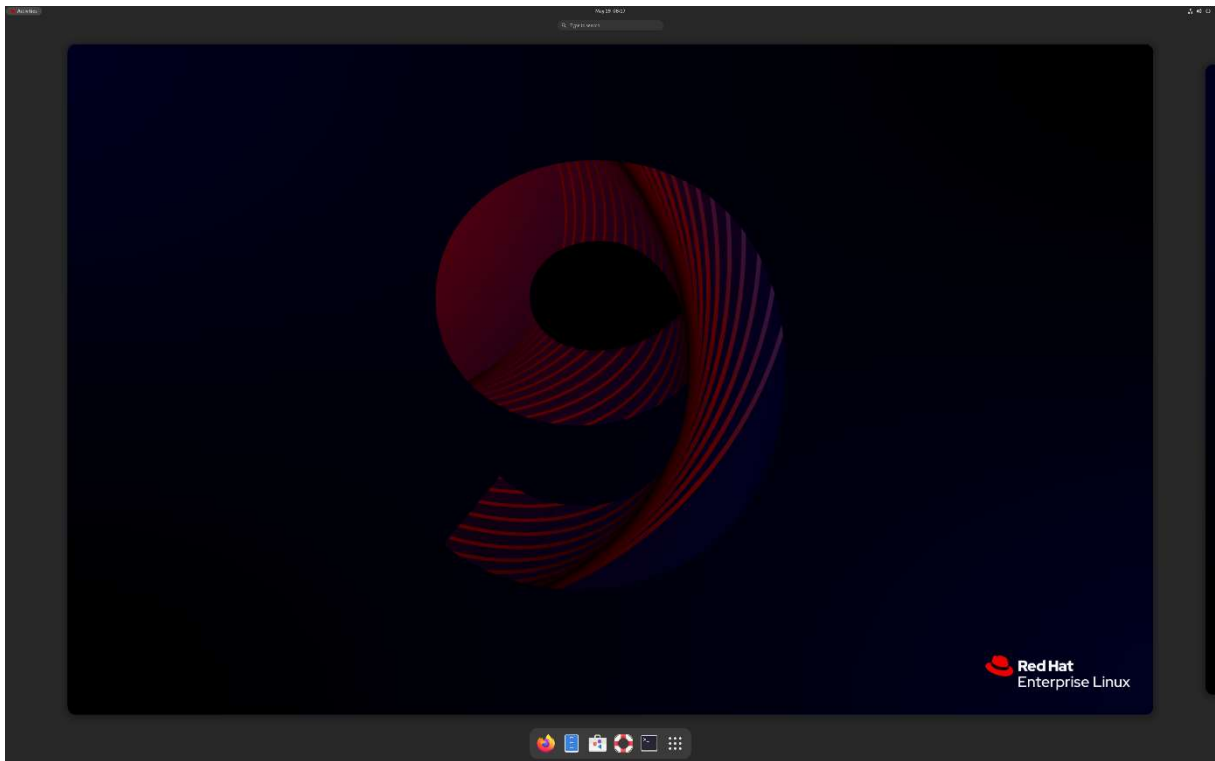
elementary OS

elementary OS est une distribution Linux qui se distingue par son esthétique élégante et son interface utilisateur soignée. Basée sur Ubuntu, cette distribution met l'accent sur la simplicité, la convivialité et l'expérience utilisateur.



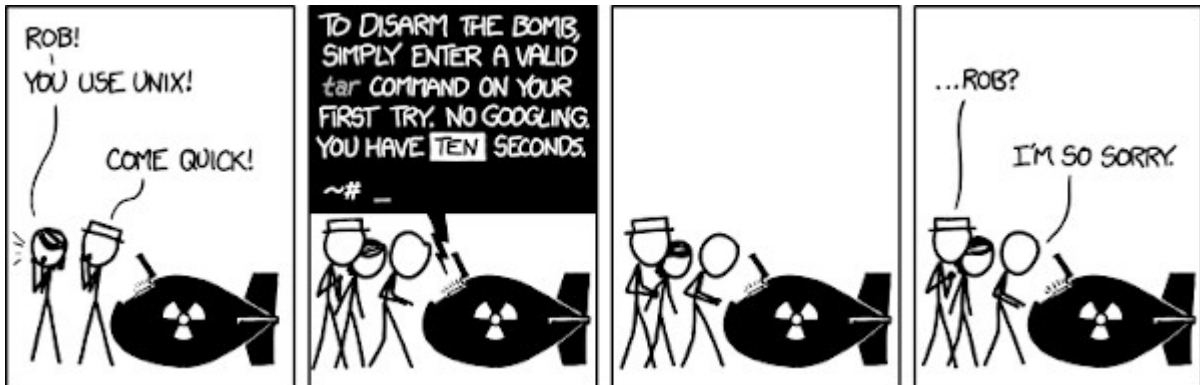
Rocky Linux / CentOS / Red Hat Enterprise Linux / AlmaLinux

Red Hat est une entreprise renommée dans le monde des logiciels open source, connue pour son système d'exploitation Red Hat Enterprise Linux (RHEL) et ses services de support professionnel. En plus de Red Hat, il existe de nombreuses distributions Linux alternatives qui sont également largement utilisées dans le monde de l'entreprise. Parmi les principales alternatives, on trouve CentOS, qui était autrefois une version gratuite et compatible avec RHEL avant d'être transformée en CentOS Stream. Afin de continuer le travail de CentOS consistant à créer une copie parfaite et sans licence commerciale de Red Hat, Rocky Linux et AlmaLinux ont vu le jour.



La ligne de commande

La ligne de commande Linux, souvent appelée "terminal", est un outil essentiel qui permet aux utilisateurs d'interagir directement avec le système d'exploitation en saisissant des commandes textuelles. Bien que les interfaces graphiques (GUI) aient rendu l'informatique plus accessible, la ligne de commande reste un moyen puissant et efficace de gérer, de contrôler et de diagnostiquer un système Linux. Dans cette page, nous explorerons les bases de la ligne de commande Linux, montrant comment elle fonctionne, comment naviguer dans le système de fichiers, et comment effectuer des tâches courantes.



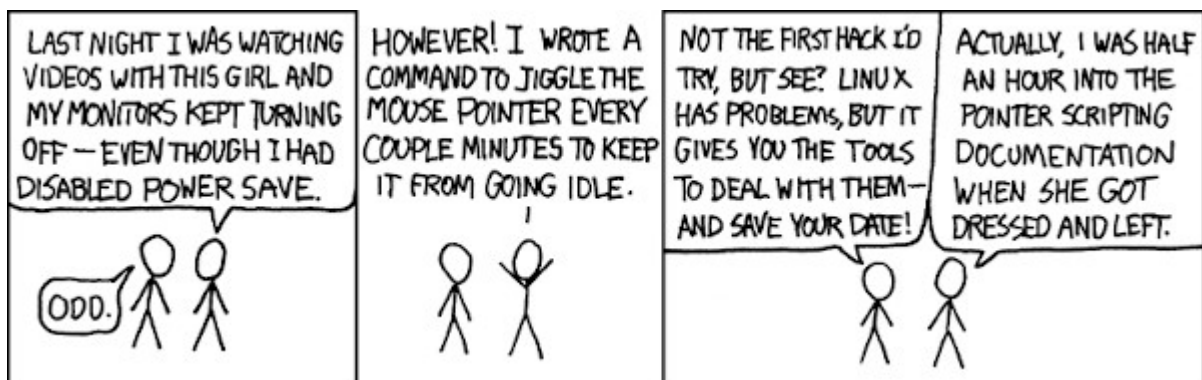
La Structure de la Ligne de Commande

La ligne de commande Linux est généralement composée de trois éléments principaux :

Le Prompt: C'est le symbole affiché par le terminal, qui indique que le système est prêt à recevoir une commande. Il peut prendre diverses formes, mais le plus courant est le "\$" pour un utilisateur ordinaire et le "#" pour le superutilisateur (root).

La Commande: C'est l'instruction que l'utilisateur souhaite exécuter. Par exemple, "ls" pour lister les fichiers ou "mkdir" pour créer un répertoire.

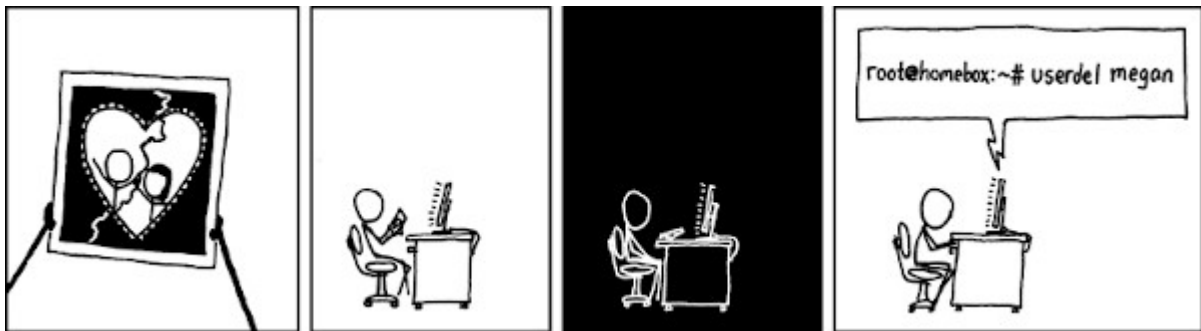
Les Arguments: Ce sont des options ou des paramètres spécifiques à la commande. Par exemple, "ls -l" affiche les fichiers sous forme de liste détaillée. Commandes de base (ls, cd, pwd, mkdir, touch, cp, mv, rm)



Gestion des utilisateurs et des permissions

Création d'utilisateurs

La création d'utilisateurs sous Linux est une procédure courante qui permet de définir des identités individuelles sur un système. Pour ce faire, les administrateurs système utilisent généralement la commande "useradd". Cette commande permet de créer un nouvel utilisateur avec un nom spécifié, un identifiant d'utilisateur unique (UID), un répertoire personnel, et un shell par défaut. Une fois l'utilisateur créé, il est possible de définir un mot de passe pour sécuriser l'accès à son compte en utilisant la commande "passwd". La gestion des utilisateurs sous Linux est essentielle pour gérer l'accès aux ressources du système, contrôler les permissions et garantir la sécurité.



Attribution de permissions (chmod, chown)

L'attribution de permissions, également appelée gestion des permissions, est un processus essentiel sous Linux qui permet de déterminer qui peut accéder à quelles ressources et quelles actions sont autorisées. Chaque fichier et répertoire sur un système Linux est associé à un ensemble de permissions qui spécifient les droits d'accès pour différents utilisateurs et groupes. Ces permissions peuvent inclure la possibilité de lire, écrire, exécuter, ou même de supprimer des fichiers. Les commandes "chmod" et "chown" sont couramment utilisées pour modifier les permissions et propriétaires de fichiers et de répertoires. La gestion minutieuse des permissions est cruciale pour la sécurité et la confidentialité des données, car elle permet de contrôler l'accès aux informations sensibles et de prévenir les modifications non autorisées.

Comptes superutilisateur (root)

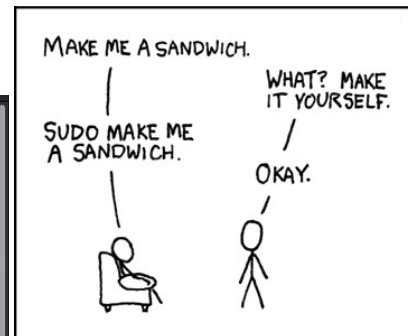
Le compte utilisateur "root" est l'utilisateur superutilisateur ou administrateur le plus puissant sur un système Linux. Connu sous le nom de "superuser", le compte "root" a un accès complet à toutes les ressources du système et peut effectuer n'importe quelle action, y compris la modification ou la suppression de fichiers système critiques. Cependant, avec un grand pouvoir vient une grande responsabilité, car l'utilisation imprudente du compte "root" peut entraîner des dommages graves au système. Par conséquent, l'accès au compte "root" est généralement restreint et limité aux tâches d'administration essentielles. Les administrateurs système sont encouragés à utiliser le compte "root" avec précaution et à privilégier l'utilisation de comptes utilisateur standard pour la plupart des tâches, afin de minimiser les risques d'erreurs catastrophiques.


```
myuser@debian1:/home/linuxhint$ sudo apt update

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for myuser:
myuser is not in the sudoers file. This incident will be reported.
myuser@debian1:/home/linuxhint$
```



Gestion des fichiers et des répertoires

Structure du système de fichiers Linux

La structure du système de fichiers Linux est organisée de manière hiérarchique, ce qui reflète la façon dont les fichiers et les répertoires sont organisés sur un système. Au sommet de cette hiérarchie se trouve le répertoire racine, représenté par `/`. Sous le répertoire racine, se trouvent une série de répertoires principaux, tels que `/home` pour les répertoires personnels des utilisateurs, `/etc` pour les fichiers de configuration système, et `/bin` pour les binaires système essentiels. Les répertoires et les fichiers sont organisés de manière logique pour faciliter la gestion et l'accès aux ressources du système. Cette structure hiérarchique permet également de maintenir l'ordre et la cohérence sur le système, ce qui est essentiel pour le bon fonctionnement de Linux. Les utilisateurs et les administrateurs peuvent naviguer dans cette structure en utilisant des commandes de ligne de commande telles que `cd` pour changer de répertoire et `ls` pour lister les fichiers et les répertoires.

Navigation dans les répertoires

La navigation dans les répertoires sous Linux est une compétence fondamentale pour travailler efficacement avec le système d'exploitation. Pour parcourir la hiérarchie des répertoires, les utilisateurs et les administrateurs utilisent principalement la commande `cd` (changement de répertoire) dans le terminal. Par exemple, `cd /home/utilisateur` déplace le répertoire de travail vers le répertoire personnel de l'utilisateur. La commande `cd ..` permet de remonter d'un niveau dans la hiérarchie, tandis que `cd` sans argument revient au répertoire personnel de l'utilisateur. La commande `pwd` (afficher le répertoire de travail) permet de connaître le répertoire actuel. Pour naviguer rapidement, des raccourcis tels que `~` pour le répertoire personnel de l'utilisateur sont disponibles. En maîtrisant ces commandes de base, les utilisateurs peuvent facilement se déplacer dans le système de fichiers Linux et accéder aux fichiers et aux répertoires nécessaires pour leurs tâches.

Manipulation de fichiers et de répertoires (cp, mv, rm)

La manipulation de fichiers et de répertoires sous Linux est une compétence essentielle pour gérer efficacement les données et les ressources du système. Les commandes de base telles que `ls` (list) permettent de répertorier les fichiers et les répertoires dans un répertoire donné, tandis que `cp` (copy) permet de copier des fichiers et des répertoires, et `mv` (move) permet de déplacer ou de renommer des fichiers. La commande `rm` (remove) est utilisée pour supprimer des fichiers et des répertoires, mais elle doit être utilisée avec prudence pour éviter la perte de données.

L'utilitaire a tout faire : mc

Midnight Commander, souvent abrégé en `mc`, est un puissant utilitaire de gestion de fichiers en mode texte pour les systèmes Unix, y compris Linux. Il offre une interface utilisateur en ligne de

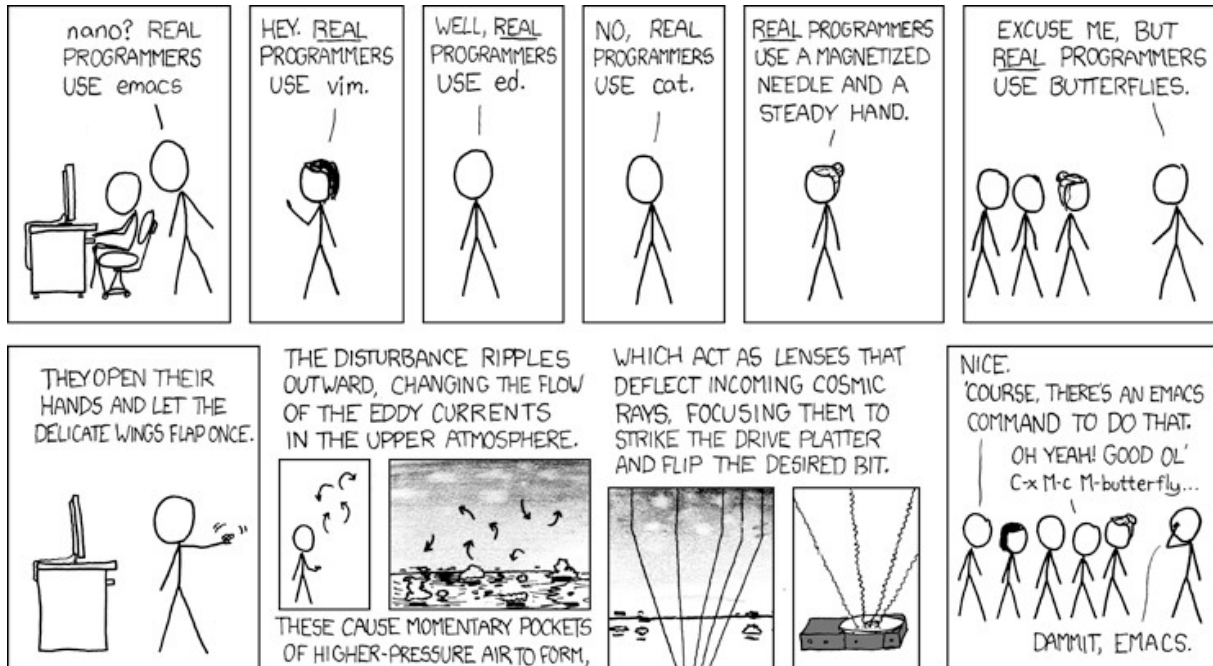
commande à deux panneaux, similaire à l'Explorateur de fichiers sous Windows, ce qui permet aux utilisateurs de naviguer rapidement dans le système de fichiers, de copier, de déplacer, de renommer, de compresser et de visualiser des fichiers, le tout à partir du terminal. Midnight Commander propose également une gamme d'autres fonctionnalités utiles, notamment l'édition de fichiers, l'accès aux archives, la gestion des permissions et la personnalisation de l'interface. En raison de sa polyvalence et de sa facilité d'utilisation, Midnight Commander est très apprécié des utilisateurs avancés et des administrateurs système pour la gestion de fichiers sur des systèmes Linux en mode texte.

Left	File	Command	Options	Right			
<- />				<- ~/pdfedit/src/gui>			
Name		Size	MTime	Name		Size	MTime
/.mc		1024	Apr 24 01:24	selfdest~dget.cc		971	May 4 15:13
/bin		3072	Jun 1 02:24	selfdest~idget.h		679	Jul 18 02:08
/boot		1024	Apr 24 18:16	settings.cc		12308	Jul 18 02:08
~cdrom		11	Apr 23 19:56	settings.h		2061	Jul 18 02:08
/dev		29696	Jul 25 07:37	staticse~ings.cc		3283	Jun 10 19:05
/etc		3072	Jul 25 07:36	staticsettings.h		643	Jun 10 03:51
/home		1024	Apr 23 20:25	statusbar.cc		2148	Jun 6 04:19
/initrd		1024	Apr 23 19:58	statusbar.h		959	Jun 12 04:44
/lib		5120	Jun 1 02:27	stringoption.cc		2015	Jul 18 02:08
~lib64		4	May 31 09:24	stringoption.h		837	Jul 18 02:08
/lost+found		12288	Apr 23 19:55	stringpr~erty.cc		2670	Jul 18 02:08
/media		1024	Apr 23 19:56	stringproperty.h		1025	Jul 18 02:08
/mnt		1024	Apr 24 18:23	test.qs		1454	Jun 24 02:11
/opt		1024	Apr 23 19:58	toolbar.cc		2141	Jul 18 02:08
/proc		0	Jul 25 07:36	toolbar.h		1113	Jul 18 02:08
/root		1024	Apr 24 20:12	toolbutton.cc		1931	Jul 18 02:08
/sbin		3072	Jun 1 02:25	toolbutton.h		1214	Jul 18 02:08
-> /lib				staticsettings.h			
bilbo@u64:~/pdfedit/src/gui\$				[^]			
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDm 10Quit							

Éditeurs de texte

La guerre des éditeurs de texte

La "guerre des éditeurs de texte" entre les utilisateurs de vi et les utilisateurs d'Emacs est une longue tradition au sein de la communauté Linux et des développeurs Unix. Vi, représentant une approche minimaliste avec une interface en mode texte, et Emacs, offrant un environnement plus complexe avec de nombreuses fonctionnalités, ont chacun leur base d'adeptes passionnés. Cette rivalité a donné lieu à des débats animés sur les forums et dans les communautés de développement, chacun camp défendant son éditeur de texte préféré.



Emacs

Emacs est un éditeur de texte puissant et extensible qui a une place spéciale dans le monde de la programmation et de l'informatique depuis des décennies. Emacs est également célèbre pour son utilisation de commandes basées sur des combinaisons de touches. En plus de son rôle en tant qu'éditeur de texte, Emacs peut être utilisé pour la gestion de tâches, la navigation sur le web, la messagerie électronique.

```
File Edit Options Buffers Tools Help
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Get help          C-h (Hold down CTRL and press h)
Emacs manual      C-h r      Browse manuals    C-h i
Emacs tutorial    C-h t      Undo changes     C-x u
Buy manuals       C-h RET    Exit Emacs       C-x C-c
Activate menubar  M-`

('C-' means use the CTRL key. 'M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)
Useful tasks:
Visit New File    Open Home Directory
Customize Startup Open *scratch* buffer

GNU Emacs 27.1 (build 1, x86_64-pc-linux-gnu, GTK+ Version 3.24.30, cairo version 1.16.0)
of 2022-01-24, modified by Debian
Copyright (C) 2020 Free Software Foundation, Inc.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software--Free as in Freedom--so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-o for information on getting the latest version.
```

Vim

Vim, acronyme de "Vi IMproved", est un éditeur de texte en mode texte largement utilisé dans le monde Unix et Linux. Vim est apprécié pour sa vitesse, son efficacité et sa polyvalence. Il offre un large éventail de fonctionnalités pour l'édition de texte, la navigation et la gestion des fichiers. Vim se caractérise par sa conception basée sur des modes, ce qui signifie que l'utilisateur peut basculer entre différents modes pour effectuer différentes actions, telles que l'édition, la navigation ou la suppression de texte.

```
VIM - Vi IMproved

version 7.0.235
by Bram Moolenaar et al.
Vim is open source and freely distributable


Help poor children in Uganda!

type :help iccf<Enter>      for information

type :q<Enter>              to exit
type :help<Enter> or <F1>   for on-line help
type :help version7<Enter> for version info
```

0,0-1 All

Nano

Nano est un éditeur de texte simple et convivial qui s'adresse principalement aux utilisateurs débutants ou à ceux qui préfèrent une interface plus intuitive. Contrairement aux éditeurs de texte en mode texte comme Vim, Nano offre une interface utilisateur basée sur des menus et des raccourcis clavier faciles à mémoriser.

```

GNU nano 4.2                                     README                                     Modified
1
2          GNU nano -- a simple editor, inspired by Pico
3
4 Overview
5
6   The nano project was started because of a few "problems" with the
7   wonderfully easy-to-use and friendly Pico text editor.
8
9   First and foremost was its license: the Pine suite does not use
10  the GPL, and (before using the Apache License) it had unclear
11  restrictions on redistribution.  Because of this, Pine and Pico
12  were not included in many GNU/Linux distributions.  Furthermore,
13  some features (like go-to-line-number or search-and-replace) were
14  unavailable for a long time or require a command-line flag.  Yuck.
15
16  Nano aimed to solve these problems by: 1) being truly free software
17  by using the GPL, 2) emulating the functionality of Pico as closely
18  as is reasonable, and 3) include extra functionality by default.
19
20  Nowadays, nano wants to be a generally useful editor, with default
21  settings that do not change the file contents automatically.
22
23  [ Justified selection ]
24
25  ^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
26  ^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line

```

Gestion des packages et des logiciels

Gestionnaires de paquets (apt, yum)

Les gestionnaires de paquets sont des outils logiciels essentiels dans les systèmes d'exploitation basés sur Linux, et ils simplifient grandement la gestion et l'installation de logiciels. Ils permettent aux utilisateurs de rechercher, d'installer, de mettre à jour et de supprimer des logiciels de manière transparente. Parmi les gestionnaires de paquets les plus couramment utilisés, on trouve APT (Advanced Package Tool) pour les distributions Debian et Ubuntu, YUM (Yellowdog Updater, Modified) pour les distributions basées sur Red Hat, et Pacman pour Arch Linux. Ces gestionnaires de paquets utilisent des référentiels de logiciels pour fournir un accès à une vaste bibliothèque de programmes open source. En plus de simplifier la gestion des logiciels, les gestionnaires de paquets garantissent que les dépendances entre les logiciels sont gérées de manière cohérente, ce qui contribue à la stabilité du système. Ils jouent un rôle clé dans la gestion des mises à jour de sécurité, ce qui est essentiel pour maintenir un système Linux sécurisé et à jour.

Installation et mise à jour de logiciels

Debian, Mint, Ubuntu : APT

Pour installer, mettre à jour et désinstaller des logiciels avec APT (Advanced Package Tool) sur les distributions basées sur Debian, vous pouvez utiliser les commandes suivantes :

Installation : Pour installer un logiciel, utilisez la commande `sudo apt install nom_du_paquet`. Par exemple, pour installer Firefox, vous pouvez saisir `sudo apt install firefox`. APT résoudra automatiquement les dépendances nécessaires et téléchargera puis installera le logiciel.

Mise à jour : Pour mettre à jour la liste des paquets disponibles et mettre à jour les logiciels déjà installés, utilisez `sudo apt update` suivi de `sudo apt upgrade`. La première commande met à jour la liste des paquets disponibles, tandis que la seconde installe les mises à jour disponibles.

Désinstallation : Pour désinstaller un logiciel, utilisez `sudo apt remove nom_du_paquet`. Par exemple, pour désinstaller Firefox, vous pouvez saisir `sudo apt remove firefox`. Si vous souhaitez également supprimer les fichiers de configuration associés, utilisez `sudo apt purge nom_du_paquet`.

Red Hat, CentOS, Rocky Linux, AlmaLinux : yum

Pour installer, mettre à jour et désinstaller des logiciels avec YUM (Yellowdog Updater, Modified) sur les distributions basées sur Red Hat, comme CentOS ou Fedora, vous pouvez utiliser les commandes suivantes :

Installation : Pour installer un logiciel, utilisez la commande `sudo yum install nom_du_paquet`. Par exemple, pour installer le navigateur web Firefox, vous pouvez saisir `sudo yum install firefox`. YUM gèrera automatiquement les dépendances et téléchargera puis installera le logiciel.

Mise à jour : Pour mettre à jour la liste des paquets disponibles et mettre à jour les logiciels déjà installés, exécutez `sudo yum check-update` suivi de `sudo yum update`. La première commande vérifie les mises à jour disponibles, tandis que la seconde les installe.

Désinstallation : Pour désinstaller un logiciel, utilisez `sudo yum remove nom_du_paquet`. Par exemple, pour désinstaller Firefox, vous pouvez saisir `sudo yum remove firefox`.

SUSE, OpenSUSE : YaST

Pour installer, mettre à jour et désinstaller des logiciels en ligne de commande avec YaST, vous pouvez utiliser les commandes suivantes :

Installation : Pour installer un logiciel, utilisez la commande `sudo zypper install nom_du_paquet`. Par exemple, pour installer le navigateur web Firefox, vous pouvez saisir `sudo zypper install firefox`. YaST gèrera automatiquement les dépendances et installera le logiciel.

Mise à jour : Pour mettre à jour la liste des paquets disponibles et mettre à jour les logiciels déjà installés, exécutez `sudo zypper refresh` suivi de `sudo zypper update`. La première commande met à jour la liste des paquets disponibles, tandis que la seconde les installe.

Désinstallation : Pour désinstaller un logiciel, utilisez la commande `sudo zypper remove nom_du_paquet`. Par exemple, pour désinstaller Firefox, vous pouvez saisir `sudo zypper remove firefox`.

Gestion des processus

Affichage des processus en cours d'exécution (ps, top)

Pour afficher les processus en cours d'exécution à l'aide de la commande "ps" ou de l'utilitaire "top" dans un terminal Linux, vous avez deux options puissantes. Pour "ps", la commande de base "ps aux" est couramment utilisée pour afficher une liste complète des processus avec des informations détaillées, y compris le PID, le propriétaire, l'utilisation de la CPU, la mémoire, le temps d'exécution, et bien plus encore.

```
[linuxtechnic@webserver ~]$ ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	1.7	178824	13668	?	Ss	Jan19	0:05	/usr/lib/systemd/systemd --swi
root	2	0.0	0.0	0	0	?	S	Jan19	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	Jan19	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	Jan19	0:00	[rcu_par_gp]
root	6	0.0	0.0	0	0	?	I<	Jan19	0:00	[kworker/0:0H]
root	8	0.0	0.0	0	0	?	I<	Jan19	0:00	[mm_percpu_wq]
root	9	0.0	0.0	0	0	?	S	Jan19	0:00	[ksoftirqd/0]
root	10	0.0	0.0	0	0	?	I	Jan19	0:00	[rcu_sched]
root	11	0.0	0.0	0	0	?	S	Jan19	0:00	[migration/0]
root	12	0.0	0.0	0	0	?	S	Jan19	0:00	[watchdog/0]
root	13	0.0	0.0	0	0	?	S	Jan19	0:00	[cpuhp/0]
root	15	0.0	0.0	0	0	?	S	Jan19	0:00	[kdevtmpfs]
root	16	0.0	0.0	0	0	?	I<	Jan19	0:00	[netns]
root	17	0.0	0.0	0	0	?	S	Jan19	0:00	[kauditd]
root	18	0.0	0.0	0	0	?	S	Jan19	0:00	[khungtaskd]
root	19	0.0	0.0	0	0	?	S	Jan19	0:00	[oom_reaper]
root	20	0.0	0.0	0	0	?	I<	Jan19	0:00	[writeback]
root	21	0.0	0.0	0	0	?	S	Jan19	0:00	[kcompactd0]
root	22	0.0	0.0	0	0	?	SN	Jan19	0:00	[ksmd]
root	23	0.0	0.0	0	0	?	SN	Jan19	0:00	[khugepaged]
root	24	0.0	0.0	0	0	?	I<	Jan19	0:00	[crypto]
root	25	0.0	0.0	0	0	?	I<	Jan19	0:00	[kintegrityd]
root	26	0.0	0.0	0	0	?	I<	Jan19	0:00	[kblockd]
root	27	0.0	0.0	0	0	?	I<	Jan19	0:00	[md]
root	28	0.0	0.0	0	0	?	I<	Jan19	0:00	[edac_poller]

Cours d'Introduction à Linux

Pour "top", il vous suffit de saisir "top" dans le terminal pour obtenir une vue en temps réel des processus actifs, triés par utilisation de la CPU. "top" met à jour automatiquement la liste toutes les quelques secondes, vous permettant de surveiller les performances du système en temps réel. Les deux outils sont essentiels pour diagnostiquer les problèmes, surveiller les ressources système et gérer les processus en cours d'exécution sur un système Linux. Pour quitter "top", appuyez sur la touche "q".

```
top - 21:52:55 up 8:13, 6 users, load average: 1.27, 1.46, 1.54
Tasks: 320 total, 1 running, 316 sleeping, 3 stopped, 0 zombie
%Cpu(s): 1.8 us, 1.4 sy, 0.0 ni, 96.6 id, 0.0 wa, 0.2 hi, 0.1 si, 0.0 st
MiB Mem : 7496.1 total, 279.4 free, 5290.3 used, 1926.4 buff/cache
MiB Swap: 7629.0 total, 7566.0 free, 63.0 used, 1424.8 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
802	kousekip	-2	0	2835368	228996	142672	S	7.6	3.0	25:33.97	kwin_wayland
6247	kousekip	20	0	7236920	1.3g	730088	S	3.6	18.2	79:09.18	GeckoMain
19115	root	20	0	0	0	0	D	2.6	0.0	0:12.39	kworker/u32:8+events_unbound
15852	root	20	0	0	0	0	I	2.3	0.0	1:34.28	kworker/u32:6+events_unbound
6440	kousekip	20	0	3011096	274800	137616	S	1.7	3.6	1:38.78	Web Content
14574	kousekip	20	0	49.6g	613388	169892	S	1.7	8.0	14:45.06	Web Content
917	kousekip	20	0	4687960	291956	15496	S	1.3	3.8	8:02.84	plasmashell
16147	kousekip	20	0	66.8g	385516	161276	S	1.3	5.0	5:16.05	DiscordCanary
7516	kousekip	20	0	271316	52104	33780	S	1.0	0.7	5:36.12	GeckoMain
19069	kousekip	20	0	1980648	156952	100456	S	1.0	2.0	0:12.06	spectacle
19463	root	20	0	0	0	0	I	1.0	0.0	0:13.81	kworker/u32:9-phy0
1179	kousekip	20	0	1338048	114616	88384	S	0.7	1.5	1:13.89	konsole
6627	kousekip	20	0	32.7g	307008	111496	S	0.7	4.0	13:18.66	WebExtensions
15329	kousekip	20	0	3239172	396072	159600	S	0.7	5.2	8:47.12	Web Content
19807	kousekip	20	0	12156	4328	3492	R	0.7	0.1	0:00.66	top
165	root	-2	0	0	0	0	S	0.3	0.0	1:53.89	gfx
428	systemd+	20	0	14344	4724	3984	S	0.3	0.1	2:03.05	systemd-oomd
457	root	-51	0	0	0	0	S	0.3	0.0	0:57.13	irq/78-rtw88_pc
518	root	20	0	1873016	25636	6968	S	0.3	0.3	0:06.08	python
2219	kousekip	20	0	364572	35320	27744	S	0.3	0.5	0:02.09	xdg-desktop-por
7127	kousekip	20	0	3635240	384700	170576	S	0.3	5.0	24:04.39	Web Content
18741	kousekip	20	0	3370488	452268	201524	S	0.3	5.9	2:52.37	Web Content
1	root	20	0	165600	9556	6556	S	0.0	0.1	0:03.89	systemd

Un logiciel optionnel (glances) vous permettra d'avoir une vue plus détaillée sur l'ensemble du système

```
www.tastogest.co.uk (CentOS Linux 8.1.1911_44bit / Linux 4.18.0-147.5.1.el8_1.x86_64) Uptime: 3 days, 14:04:55
```

CPU	MEM	SWAP	LOAD	8-core
100%	59.4%	6.8%	1.27	0.05
100%	59.4%	6.8%	1.46	0.05
100%	59.4%	6.8%	1.54	0.05

NETWORK	Rx/s	Tx/s	TASKS	157 (200 tch), 1 run, 79 smp, 77 oth sorted automatically by CPU consumption
eth0	280b	280b		
lo	280b	280b		
TCP CONNECTIONS				
Listen	4	0.0	1.0	150M 10.3M 10408 root
Initiated	1	0.0	0.0	91.1M 9.27M 10413 ansible
Established	1	0.0	0.0	41M 9.13M 987 root
Terminated	1	0.0	0.0	23M 7.91M 1 root
Tracked	2/24576	0.0	0.0	215M 5.35M 971 root
DISK I/O	R/s	W/s		
dm-0	0	0	0.4	108M 4.20M 759 root
dm-1	0	0	0.4	101M 3.93M 762 root
loop0	0	0	0.4	30M 3.74M 10417 ansible
loop1	0	0	0.4	30M 3.73M 981 root
sd	0	0	0.4	93.1M 3.59M 972 root
sdal	0	0	0.3	79.8M 3.19M 951 dbus
sdsl	0	0	0.3	213M 2.93M 960 root
sr0	0	0	0.2	205M 2.30M 1434 root
FILE SYS	Used	Total		
/ (loop0)	94.0M	94.0M	0.0	1.11M 1.49M 1429 systemd-s
/etc (cl-root)	1.77G	9.99G	0.0	248M 1.44M 970 root
/etc/naswitch.conf	94.0M	94.0M	0.0	73.6M 994K 926 root
/etc/rsi (loop0)	94.0M	94.0M	0.0	122M 992K 950 root
/home (cl-root)	1.77G	9.99G	0.0	90.1M 548K 999 root
/lib/firmware	1.77G	9.99G	0.0	204M 488K 949 root
/lib/modules (cl-root)	1.77G	9.99G	0.0	32.6M 212K 985 root
/usr (cl-root)	1.77G	9.99G	0.0	64.4M 128K 952 root
/root (cl-root)	1.77G	9.99G	0.0	0 0 2 root
/snap (cl-root)	1.77G	9.99G	0.0	0 0 3 root
/snap/core/9066	94.0M	94.0M	0.0	0 0 4 root
/snap/glances/517	1.77G	9.99G	0.0	0 0 6 root
/tmp (cl-root)	1.77G	9.99G	0.0	0 0 8 root
/tmp (cl-root)	1.77G	9.99G	0.0	0 0 9 root
/usr/src (cl-root)	1.77G	9.99G	0.0	0 0 10 root
/var/lib/snapd	1.77G	9.99G	0.0	0 0 11 root
/var/lib/snapd/hosts	1.77G	9.99G	0.0	0 0 12 root
/var/lib/snapd/hosts	1.77G	9.99G	0.0	0 0 13 root
/lib/snapd/hosts/9066	94.0M	94.0M	0.0	0 0 14 root
/snapd/snap/core/9066	94.0M	94.0M	0.0	0 0 15 root
/snapd/snap/glances/517	1.77G	9.99G	0.0	0 0 16 root
/snapd/snap/core/9066	94.0M	94.0M	0.0	0 0 17 root
/snapd/snap/glances/517	1.77G	9.99G	0.0	0 0 18 root
/var/log (cl-root)	1.77G	9.99G	0.0	0 0 20 root
/var/snap (cl-root)	1.77G	9.99G	0.0	0 0 21 root
/var/tmp (cl-root)	1.77G	9.99G	0.0	0 0 22 root
			0.0	0 0 23 root

2020-05-23 21:14:48 BST

Gestion des processus (kill)

La gestion des processus avec les commandes "kill" et "killall" est essentielle pour contrôler les processus en cours d'exécution sur un système Linux. La commande "kill" permet de terminer un processus en utilisant son numéro d'identifiant de processus (PID) spécifique. Par exemple, "kill 1234" enverra un signal pour demander au processus avec le PID 1234 de se terminer. Le signal par défaut est TERM, qui permet au processus de se terminer proprement.

D'un autre côté, "killall" offre une approche différente en permettant de tuer des processus en fonction de leur nom. Vous pouvez spécifier le nom du processus avec "killall" suivi de l'option "-e" pour tuer tous les processus portant ce nom. Par exemple, "killall -e nom_processus" mettra fin à tous les processus portant le nom "nom_processus".

Ces commandes sont particulièrement utiles pour gérer les processus qui ne répondent plus ou qui consomment trop de ressources système. Cependant, il convient d'utiliser "kill" et "killall" avec prudence pour éviter de mettre fin à des processus essentiels au bon fonctionnement du système.

Automatisation des tâches avec les scripts Shell

Introduction aux scripts Shell

Les scripts shell sont des programmes informatiques écrits dans un langage de script interprété, tels que Bash (Bourne-Again Shell), qui est couramment utilisé sur les systèmes Linux et Unix. Ils permettent d'automatiser des tâches en utilisant une séquence de commandes qui peuvent être exécutées de manière séquentielle ou conditionnelle. Les scripts shell sont polyvalents et peuvent être utilisés pour effectuer une grande variété de tâches, de la gestion de fichiers et de répertoires à l'automatisation de processus système complexes. Ils sont largement utilisés par les administrateurs système et les développeurs pour simplifier les opérations répétitives, améliorer l'efficacité du flux de travail et automatiser des tâches fastidieuses. Les scripts shell constituent un élément fondamental de l'écosystème Linux et Unix, offrant un moyen puissant et flexible de travailler avec ces systèmes d'exploitation.

Exécution de scripts

L'exécution de scripts shell est un processus fondamental dans la gestion et l'automatisation des tâches sous les systèmes Linux et Unix. Pour exécuter un script shell, vous devez d'abord créer un fichier texte contenant les commandes shell, généralement avec l'extension ".sh". Ensuite, vous devez rendre le script exécutable en utilisant la commande "chmod +x nom_du_script.sh". Une fois cela fait, vous pouvez exécuter le script en tapant "./nom_du_script.sh" dans un terminal. Les scripts shell sont largement utilisés pour automatiser des tâches courantes, effectuer des sauvegardes, planifier des opérations système, et bien plus encore. Ils offrent un moyen puissant et flexible d'interagir avec le système d'exploitation et de simplifier la gestion des systèmes Linux et Unix, ce qui en fait un outil indispensable pour les administrateurs système et les utilisateurs avancés.

Variables, boucles et conditions

Les variables, les boucles et les conditions sont des éléments essentiels pour la création de scripts shell puissants et flexibles. Les variables permettent de stocker des données et des valeurs, ce qui facilite la personnalisation des scripts. Les boucles, telles que les boucles "for" et "while", permettent d'itérer sur des séquences de données ou de répéter des tâches jusqu'à ce qu'une condition

spécifique soit remplie. Les structures de contrôle conditionnel, comme "if-then-else", permettent de prendre des décisions basées sur des conditions, ce qui rend les scripts capables de réagir de manière dynamique aux différentes situations. En combinant ces éléments, les scripts shell peuvent automatiser des tâches complexes, traiter des données, et effectuer des opérations conditionnelles, ce qui les rend extrêmement polyvalents pour la gestion et l'automatisation des systèmes Linux et Unix. La maîtrise de ces concepts est cruciale pour créer des scripts shell efficaces et personnalisés.

Un premier script shell

Créer votre premier script shell est une étape essentielle pour apprendre à automatiser des tâches sous Linux ou Unix. Suivez ces étapes simples pour créer votre propre script shell :

1. **Choisissez un éditeur de texte** : Tout d'abord, choisissez un éditeur de texte avec lequel vous êtes à l'aise. Vous pouvez utiliser des éditeurs de texte en ligne de commande comme "nano", ou des éditeurs de texte graphiques comme "Gedit"

2. **Créez un nouveau fichier** : Ouvrez votre éditeur de texte préféré et créez un nouveau fichier en tapant `nano mon_script.sh` (ou remplacez "nano" par le nom de votre éditeur). Assurez-vous que le fichier a l'extension ".sh" pour indiquer qu'il s'agit d'un script shell.

3. **Écrivez votre script** : Dans le fichier, commencez par indiquer le type de shell que vous souhaitez utiliser. Par exemple, pour Bash, ajoutez la ligne `#!/bin/bash`. Ensuite, écrivez les commandes du script. Par exemple, voici un script shell simple qui affiche "Bonjour, monde !" lorsque vous l'exécutez :

```
#!/bin/bash
```

```
echo "Bonjour, monde !"
```

4. **Rendez le script exécutable** : Après avoir enregistré votre script, rendez-le exécutable en utilisant la commande `chmod +x mon_script.sh`.

5. **Exécutez le script** : Pour exécuter votre script, tapez `./mon_script.sh` dans un terminal. Vous devriez voir la sortie du script, qui, dans cet exemple, est "Bonjour, monde !".

Vous venez de créer et d'exécuter votre premier script shell. À partir de là, vous pouvez explorer davantage en ajoutant des variables, des boucles et des structures de contrôle conditionnel pour automatiser des tâches plus complexes. Les scripts shell sont des outils puissants pour la gestion et l'automatisation des systèmes Linux et Unix, et ils peuvent vous faire gagner beaucoup de temps dans vos tâches quotidiennes.

Réseau sous Linux

Configuration réseau

La configuration réseau sous Linux est un processus essentiel pour établir et gérer la connectivité des systèmes Linux. Elle implique la définition des paramètres réseau tels que l'adresse IP, le masque de sous-réseau, la passerelle par défaut et les serveurs DNS, qui permettent aux ordinateurs Linux de communiquer au sein d'un réseau local ou sur Internet. Les utilisateurs et les administrateurs système ont accès à divers outils pour configurer le réseau, notamment des commandes en ligne de commande telles que "ifconfig" (remplacée par "ip" sur de nombreuses distributions récentes), "netplan" pour les systèmes basés sur Ubuntu, ou "NetworkManager" pour une gestion plus conviviale. La configuration réseau est une compétence fondamentale pour garantir la connectivité, la sécurité et l'efficacité des systèmes Linux dans un environnement réseau, que ce soit sur des postes de travail individuels ou des serveurs en entreprise.

Commandes réseau de base (ping, ip, netstat)

La commande "ip" est un outil puissant pour configurer, surveiller et gérer les interfaces réseau et les routes sur les systèmes Linux. Voici un aperçu des principales utilisations de la commande "ip" :

Afficher des informations sur les interfaces réseau : Pour afficher des informations sur toutes les interfaces réseau actuellement activées, vous pouvez utiliser la commande "*ip a*" ou "*ip addr*". Cette commande affiche des détails tels que les adresses IP, les adresses MAC et l'état des interfaces.

Activer ou désactiver une interface réseau : Vous pouvez activer ou désactiver une interface réseau à l'aide de la commande "*ip link set*". Par exemple, pour activer l'interface "eth0", vous pouvez saisir *sudo ip link set eth0 up*. Pour la désactiver, remplacez "up" par "down".

Configurer des adresses IP statiques : La commande "*ip addr*" permet également de configurer des adresses IP statiques sur une interface. Par exemple, pour définir l'adresse IP de "eth0" à "192.168.1.2" avec un masque de sous-réseau de "24" bits, vous pouvez utiliser *sudo ip addr add 192.168.1.2/24 dev eth0*

Configurer des routes statiques : La commande "*ip route*" permet de configurer des routes statiques. Par exemple, pour ajouter une route par défaut vers une passerelle avec l'adresse IP "192.168.1.1", utilisez *sudo ip route add default via 192.168.1.1*

Supprimer des adresses IP ou des routes : Pour supprimer une adresse IP ou une route, utilisez la commande "*ip addr del*" ou "*ip route del*". Par exemple, pour supprimer une adresse IP de "eth0", saisissez *sudo ip addr del 192.168.1.2/24 dev eth0*

Sécurité sous Linux

- Gestion des pare-feux (iptables, firewallld)

- Mises à jour de sécurité

- Gestion des utilisateurs et des mots de passe

Sauvegarde et restauration

- Commandes de sauvegarde (tar, rsync)

- Planification de sauvegardes

Ressources en ligne

Documentation et forums

Communautés Linux

Conclusion

Ce cours d'introduction à Linux vous a fourni les bases nécessaires pour commencer à travailler avec ce système d'exploitation. Linux est un environnement puissant qui offre de nombreuses possibilités, que vous souhaitiez l'utiliser pour le développement, l'administration système ou d'autres tâches informatiques. Pour approfondir vos connaissances, il est recommandé de pratiquer régulièrement et d'explorer les ressources en ligne disponibles pour les utilisateurs Linux.