

Introduction à GNU Linux

Abdelali SAIDI

abdelali.saidi@gmail.com

Plan

- 1 Introduction
- 2 La norme de la hiérarchie des systèmes de fichiers
- 3 Commandes de base

Plan

- 1 Introduction
- 2 La norme de la hiérarchie des systèmes de fichiers
- 3 Commandes de base

Les licences

Logiciels open source et propriétaires

Un programme peut être délivré sous plusieurs formes:

- Le code source
- La version binaire

Les logiciels open source mettent à disposition leur codes sources, par contre, les logiciels propriétaires ne délivrent que la version compilée du programme.

Les licences

Logiciels libres et propriétaires

Le logiciel libre est une philosophie créée par Richard Stallman en 1983 à cause d'une histoire de pilote de son imprimante. D'où les quatre commandements suivants qui feront aussi la base de la licence GNU GPL (General Public License).

- La liberté d'exécuter le programme pour n'importe quel objectif
- La liberté d'étudier le code source du programme
- La liberté de distribuer des copies
- La liberté d'améliorer le programme et partager cette réussite

En 1985, Richard Stallman créa GNU project (Gnu's Not Unix) pour mettre à disposition un système d'exploitation libre.

GNU Linux

Pourquoi Unix?

Unix a été créé pour répondre à quelques besoins:

- Avoir un SE disponible 24h/24
- Avoir un SE multi-tâches
 - Chose qui représente un défi pour la gestion de mémoire et l'accès au disque dur
- Avoir un SE multi-utilisateurs
 - Un vrai challenge à l'époque

GNU Linux

La naissance de GNU Linux

En 1983, Richard Stallman annonce le projet de création d'un SE qui permettra à ses utilisateurs d'utiliser un ordinateur sans avoir besoin des programmes non libres.

- En 1990, presque tous les composants du SE étaient prêts, sauf le Kernel, il y avait quelque problèmes.
- En 1991, un étudiant finlandais Linus Torvalds décida de créer un nouveau Kernel basé sur Minix
- Ce Kernel était distribué gratuitement sous le nom de Freax et deviendra Linux par la suite.
- La première version était disponible sur Internet sous une licence propriétaire.
- En décembre 1992, Richard Stallman persuada Linus Torvalds de mettre linux sous licence GPL, chose qui a permit l'existence de GNU linux.

GNU Linux

Les distributions Linux

- Grâce à la communauté GNU linux, nous avons une centaine de distributions. Ces derniers ont le même Kernel Linux et ont chacun un ensemble d'outils pour la complétion du SE en respectant le Filesystem Hierarchy Standard
- Les distributions les plus populaires sont: Ubuntu, Debian, Gentoo, Mandriva, Slackware, Suse et bien sûr Redhat

GNU Linux

Avantages et désavantages des distributions

Les avantages:

- La liberté d'installation, d'exécution et de modification

Les désavantages:

- Le manque de supports techniques, quoique, il existe beaucoup de forum pour l'aide
- Le coût de son déploiement dans les entreprises, car il demande des mains d'oeuvres qualifiées.
- Même si le GNU Linux devient de plus en plus populaire, il est inévitable de former les employés

Plan

- 1 Introduction
- 2 La norme de la hiérarchie des systèmes de fichiers
- 3 Commandes de base

Filesystem Hierarchy Standard

Présentation

- La norme est une convention qui décrit l'organisation des répertoires principaux de Linux.
- Grâce à cette norme:
 - À l'installation d'un logiciel, les fichiers seront mis dans les bons dossiers peu importe la distribution Linux.
 - Une compatibilité entre les distributions existera

La racine /

Présentation

“/” est le dossier parent de tous les dossiers. Ci-dessous, la liste de ses sous-dossiers:

- /bin: contient les commandes de base pour tous les utilisateurs
- /boot: contient le chargeur d'amorçage
- /dev: contient les fichiers qui représentent les périphériques
- /etc: contient les fichiers de configuration
- /home: contient les répertoires des utilisateurs
- /lib: contient les bibliothèques des logiciels
- /mnt: le point de montage des fichiers systèmes temporaires

La racine /

Présentation

“/” est le dossier parent de tous les dossiers. Autrement appelé: la racine. Ci-dessous, la liste de ses sous-dossiers:

- /opt: contient les logiciels optionnels
- /proc: permet l'accès direct au Kernel
- /root: dossier personnel de l'utilisateur root
- /sbin: contient les exécutables pour les administrateurs
- /tmp: contient les fichiers temporaires
- /usr: User System Resources
- /var: contient les fichiers qui sont souvent consultés (logs, mails ...)

/USR

Présentation

Ce dossier contient quelques sous dossiers semblables à ceux présents dans la racine et qui ne sont pas nécessaire pour le fonctionnement minimale du système (lecteur video, navigateur web, ...).

Structure

- /usr/bin: contient des binaires exécutables en complément de /bin
- /usr/lib: contient les bibliothèques partagées
- /usr/man: contient les pages du manuel
- /usr/sbin: contient des binaires pour l'administrateur (complément de /sbin)
- /usr/src: Sources du Kernel
- /usr/X11R6: contient les fichiers du serveur graphique

/VAR

Présentation

À cause du fait que ce dossier contient les fichiers qui changent beaucoup, il ne contiendra jamais de fichier de configuration.

Structure

- /var/log: contient tous les fichiers de journalisation
- /var/mail: contient les e-mails de l'utilisateur
- /var/run: contient les fichiers temporaires des logiciels en cours d'exécution
- /var/spool: représente une file d'attente (envoi d'e-mail, impression, crontab, ...)

Plan

- 1 Introduction
- 2 La norme de la hiérarchie des systèmes de fichiers
- 3 Commandes de base**

Liste de commandes

Syntaxe :

nomCommande [options] [arguments]

- nomCommande : ordre à passer au système d'exploitation
- options : comportement de la commande
- arguments : la cible de la commande

Liste de commandes

- `cd rep`: aller au répertoire `rep`
- `ls rep`: lister le contenu de `rep`
- `cp src dest`: copier le fichier source à un fichier destination
- `mv src dest`: déplacer le fichier source à un destination
- `mkdir rep`: créer le dossier `rep`
- `rm fichier`: supprimer le fichier `rm`
- `rmdir rep`: supprimer un dossier
- `pwd`: affiche le répertoire courant

Liste de commandes

- `ln src dest`: créer un lien destination pour pointer la source
- `touch fichier`: créer un fichier vide
- `cat fichier`: voir le contenu du fichier sur la sortie standard
- `more fichier`: voir le contenu du fichier page par page
- `less fichier`: même résultat que `more` mais sans avoir à charger tout le document et contient des options de recherche
- `tail -n fichier`: affiche les `n` dernières lignes du fichier
- `head -n fichier`: affiche les `n` premières lignes du fichier

Liste de commandes

- clear : efface le terminal
- grep : cherche un mot et affiche les lignes où il se trouve
- type : trouver le type d'une commande
- cut : afficher le contenu d'un fichier en coupant des sections et sans modifier le fichier source
- wc : calcule et affiche le nombre de lignes, de mots et de caractères d'un fichier
- dmesg : affiche les logs du kernel
- whoami: affiche le nom de l'utilisateur courant

Les caractères spéciaux

Présentation

Les caractères spéciaux aident à manipuler plusieurs pattern au même moment.

* : remplace plusieurs caractères

? : remplace un seul caractère

[a-z] : remplace un caractère défini entre a et z en minuscule

[^a-z] : remplace tout caractère sauf ceux présents entre a et z en minuscule

Exemples

- `ls *.txt` : liste tous les fichiers d'extension "txt"
- `ls t*[^m]` : liste tous les fichiers qui commencent par "t" suivi d'une suite de caractères qui ne se terminent pas avec un "m"
- `ls t?st.c` :
- `ls *st*` :

Éditeur de texte

VI Editor

Présentation

VI est un éditeur de texte en ligne de commande. Il est l'éditeur par défaut sur la famille Debian.

Lancer VI

Pour éditer un fichier texte existant ou bien en créer un nouveau, lancez la commande :

```
$ vi fichier
```

Modes d'utilisation

VI n'est pas un éditeur de texte "WYSIWYG". On l'utilise par modes:

- Mode d'insertion: ici on tape notre texte (on y accède en tapant i)
- Mode de commande: ici on manipule notre texte ou bien notre fichier (on y accède en tapant ESC)

Éditeur de texte

VI Editor

Mode de commande

- `:w` : enregistre le document
- `:q` : quitte le document
- `:q!` : quitte le document sans faire d'enregistrement
- `:wq` ou `:x` : enregistre et quitte
- `:w fich` : enregistre le fichier avec un nouveau nom
- `/mot` : cherche "mot" dans le document. On tape "n" pour passer à une occurrence nouvelle
- `v` : le mode visuel pour faire une sélection de texte
- `d/dd` : couper/couper une ligne entière
- `y/yy` : copier/copier une ligne entière
- `p` : coller

Éditeur de texte

VI Editor

Mode d'insertion

- *a* : ajouter du texte après le curseur
- *A* : ajouter du texte à la fin d'une ligne
- *i* : insérer du texte avant le curseur
- *I* : insérer du texte au début d'une ligne
- *o* : ajouter une nouvelle ligne après la position actuelle du curseur
- *O* : ajouter une nouvelle ligne avant la position actuelle du curseur

Archiver et compresser

La commande tar

Présentation

La commande *tar* est une vieille commande qui sert à archiver des fichiers (les mettre en un seul fichier). Par default, cette commande ne fait pas de compression. La syntaxe:

```
$ tar [options][fichiers]
```

Options:

- -x : extraire le contenu d'un fichier
- -c : créer un nouveau archive
- -t : afficher le contenu d'un archive sans extraction
- -f fich : spécifier le fichier à utiliser
- -v : le mode verbose
- -z : compresser ou décompresser en utilisant la commande gzip
- -p : préserver les permissions et le propriétaire
- -C : changer le dossier spécifié

Archiver et compresser

La commande gzip

Présentation

La commande *gzip* est désignée pour la compression de fichiers. *Gunzip* sert à la décompression. La syntaxe:

```
$ gzip [options][fichiers]
```

Options:

- -1 à -9 : fixer la qualité de la compression (-1 est le plus rapide, -9 apporte la meilleure compression)
- -c : ne pas remplacer le fichier d'origine et envoyer le résultat à la sortie standard
- -l : informations à propos du fichier
- -t : le mode récursif

Recherche de fichiers

La commande *find*

La commande *find* sert à rechercher les fichiers. On peut faire des recherches minutieuses grâce aux options offertes. La syntaxe:

```
$ find [path][options]
```

Options:

- -atime +n (-n) : chercher les fichiers qui ont été accédé il y a + ou - de n jours
- -mtime +n (-n) : chercher les fichiers qui ont été modifié il y a + ou - de n jours
- -name "toto*" : chercher les fichiers qui ont un nom commençant par toto
- -maxdepth n : définir la profondeur max de la recherche
- -type type : définir le type de l'objet recherché
 - f pour les fichiers
 - d pour les dossiers
 - l pour les liens symboliques

Recherche de fichiers

La commande *slocate* - *locate*

Locate et slocate ne font pas la recherche sur le système de fichier comme find, elles le font sur une base de données. Cela est bien plus rapide, seulement, il faut garder la base de données à jour (avec la commande updatedb) pour y indexer les nouveaux fichiers.

which et whereis

which et whereis sont utilisées pour découvrir la localisation d'une commande.

Tubes et redirections

Les redirections

L'objectif de la redirection est de changer le "stream" normal d'une commande. En UNIX, on trouve trois principaux streams:

- STDIN(0): le flux d'entrée (par default c'est le clavier)
- STDOUT(1): le flux de sortie (par default c'est l'écran)
- STDERR(2): le flux d'erreur (par default c'est l'écran aussi)

Exemple

- `ls /bin > contenu.txt` (crée un nouveau fichier)
- `ls /bin >> contenu.txt` (ajoute à la fin du fichier)

Tubes et redirections

Les tubes

Les tubes ont le même objectif que les redirections, seulement entre commandes.

Une première commande écrit sur le tube, la deuxième lit.

Exemple

- `ls /bin | tr a-z A-Z`

en redirection on aurait fait:

- `ls /bin > contenu.txt`
- `tr a-z A-Z < contenu.txt`