LINUX

Quelques généralités sur les manipulations sous Linux, pour des distributions dérivées de Debian (Debian, Ubuntu).

Présentation

L'utilisation de l'environnement Linux passe par la maîtrise des éléments suivants :

- l'utilisation du *shell*
- l'arborescence des fichiers du ***SGF*** (/etc, /var, ...)
- la configuration des options réseau (IP, DHCP, DNS, Routage/Passerelle)
- l'installation et la gestion du fonctionnement des ***services*** (apt / aptitude)
- l'utilisation et le paramétrage de ***fichiers de configuration***
- l'automatisation des actions par la réalisation de ***scripts***

Les lignes de codes correspondent à un environnement ***Debian/Ubuntu***.

Principes

Avant toute installation, vous devez :

- mettre à jour la liste des paquetages lors de la première utilisation de l'OS :

```
apt update
```

- mettre à jour les paquetages installés lors de la première utilisation de l'OS :

```
apt upgrade
```

- utiliser un compte système :

```
 sudo <commande> | sudo su | sudo -a | su
```

- avant tout paramétrage, faire une **copie de sauvegarde** des fichiers de configuration qui seront modifiés

```
cp <fichierAmodifier> <nomFichierSauvegarde>
```

Après toute modification, vous devez au choix

- recharger le service

```
 systemctl reload <nom_service>
// anciennes distributions : service <nom_service> reload
```

- redémarrer complètement le service, ce qui interrompt les connexions établies sur le serveur

```
systemctl restart <nom_service>
 //anciennes distributions : service <nom_service> restart
```

Utilisation du shell

Environnement shell

Sous Linux, l'utilisation de la ***ligne de commande*** se fait dans un environnement riche (le shell) qui propose des fonctions de ***complétion*** (il complète) grâce à la touche tabulation.

Mode administrateur

Les commandes d'administration qui doivent être exécutées en *mode administrateur* seront préfixées par la commande ***sudo***.

On peut aussi passer complètement en *mode super-utilisateur*

Il y a un risque car les manipulations opérées le sont sans demande de confirmation.

```
//commandes pour basculer en super-utilisateur
//(selon la distribution Linux retenue)
sudo su
sudo -a
su
```

En mode simple utilisateur, l'invite de commande se termine par un >, alors qu'en mode administrateur, l'invite se termine par un #.

Changer de compte

Il est possible de changer de compte au sein d'un terminal, par exemple pour exécuter des commandes *super-utilisateur* depuis la session d'un compte normal, ou au contraire pour tester une manipulation avec un compte normal depuis une session administrateur. On bascule vers un autre compte avec la commande :

```
su <nomCompte>
```

Lancement d'un processus autonome

Lorsqu'on lance un processus (par exemple un éditeur) depuis le shell, ce processus fils prend la main et on ne pourra utiliser le shell qu'après avoir fermé le processus fils.

On peut lancer un processus autonome déconnecté du shell en finissant la commande par & :

```
root@monLinux:/# gedit /etc/resolv.conf &
```

Aide sur une commande

On obtient de l'aide ponctuelle en utilisant l'option *–help* (ex: ***mkdir –help***) ou de l'aide détaillée en consultant le manuel avec la commande ***man <nomCommande>*** (ex: ***man chmod***).

Configuration réseau

1 Configuration de l'adressage statique (IP fixe)

La configuration de l'adresse IP, du masque ou l'activation / désactivation de la carte réseau se fait par la commande ip (***ifconfig*** pour les anciennes distributions).

```
//connaître l'IP et/ou lui affecter une IP
ip addr [add <adresse_ip>/<cidr> dev <nom_carte>]
//activer/désactiver la carte
ip link set <nom_carte> {up|down}
//versions dépréciées
#ifconfig <nom_carte> [<adresse_ip>] [netmask <valeur_masque>]
#ifconfig <nom_carte> [up|down]
```

Il est aussi possible de configurer le fichier ***/etc/network/interfaces/** ** pour garder la configuration après l'extinction de la machine, en **ajoutant** les lignes suivantes à la fin du fichier :

```
auto <nomCarte>                    # active la carte au démarrage de la
machine
iface <nomcarte> inet static       # utilise l'adressage IPv4 statique pour
cette interface
                                   # pour une carte en adressage dynamique,
on indiquera iface <nomCarte> inet dhcp
address <adresse_IP>               # adresse statique
netmask <masqueReseau>
gateway <IP_passerelle>
dns-nameservers <ip_dns-primaire> [<ip_dns_secondaire>]
```

Désactiver le parefeu

Voir sur https://www.tutodidacte.com/debian-desactiver-le-parefeu-iptables

2 Actualisation de l'adressage dynamique (DHCP)

Pour obtenir une adresse IP d'un serveur DHCP, ou pour renouveler un bail, on utilisera la commande

```
dhclient <nom_carte>
```

3 Configuration du routage

La configuration du routage (et donc de la passerelle par défaut pour un poste) utilise la commande route.

- Ajouter une ligne dans la table

```
route add <adresse_réseau> [netmask <valeur_masque>] [gateway
<adresse_passerelle>] [metric <valeur>]
```

- Ajout d'une route par défaut (donc d'une passerelle)

```
route add default [netmask <valeur_masque>] [gateway <adresse_passerelle>]
[metric <valeur>]
```

- Supprimer une ligne dans une table

```
route del <adresse_réseau> [netmask <valeur_masque>] [gateway
<adresse_passerelle>] [metric <valeur>]
```

- Afficher la table de routage

```
route –e
```

Ce paramétrage peut aussi être enregistré dans ***/etc/network/interfaces*** grâce à la clause ***gateway***.

4 Configuration du client DNS

La configuration des serveurs DNS que le client devra interroger se fait dans le fichier ***/etc/resolv.conf***

On y ajoute les adresses des serveurs sous la forme ci-contre.

Ce paramétrage peut aussi être enregistré dans ***/etc/network/interfaces*** grâce à la clause ***dns-nameservers***.

Gestion des comptes utilisateurs

Les comptes utilisateurs d'une machine Linux sont gérés de manière locale (connus de cette machine uniquement). Il est possible d'associer une gestion centralisée des comptes et de rattacher des machines à un domaine (non présenté ici).

Création d'un compte

Le compte est créé soit

- comme un compte associé à un environnement (dossier // /home/<nomCompte>/, shell perso*) par la commande //**adduser** ; cette dernière commande entre dans un dialogue pour renseigner les informations du compte (nom, prénom, téléphone, etc). Lors de l'ouverture de session, l'utilisateur sera positionné dans son dossier personnel (/*home/<nomCompte*>)

```
//compte avec dossier perso et environnement
adduser <nomCompte> [-p <motPasse>] [-g <nomGroupe>]
```

- comme élément simple (une information connue par l'OS) grâce à la commande ***useradd*** : ce compte ouvrira une session en étant positionné directement à la racine de la machine (// / //).

```
//compte sans environnement
useradd <nomCompte> [-p <motPasse>] [-g <nomGroupe>]
```

Suppression d'un compte

La suppression d'un compte s'effectuera par les commandes ***deluser*** ou ***userdel***.

```
userdel <nomCompte>
deluser <nomCompte>
```

Gestion du mot de passe

Il est possible de modifier le mot de passe d'un utilisateur grâce à la commande ***passwd***. Cette commande gère d'autres otpions comme le blocage d'un compte ou la limitation à des horaires donnés.

```
passwd <nomCompte>
```

Paquetages et services

Faire prendre en compte une langue pour l'ensemble des applications

Certaines distributions ou *appliances* fournissent l'OS uniquement configuré en anglais.

Il est possible de demander à Linux de prendre en compte d'autres langages (que les applications peuvent alors aussi utiliser.

```
dpkg-reconfigure locale //cocher les options souhaitées
```

Il peut être nécessaire de redémarrer le/les service/s devant prendre en compte ces modifications (Apache ou autre).

Gestion de paquetages

TODO : ajouter la gestion avec d'autres OS Linux (*yum* pour Fedora par exemple)

L'installation d'applications système (DNS, DHCP, serveur Web, etc) sous Linux nécessite les droits administrateur. La commande ***apt-get*** est le gestionnaire de paquetages (on pourra aussi utiliser l'équivalent ***aptitude*** ou le logiciel en interface graphique ***synaptic***).

Il permet diverses manipulations selon les options utilisées, parmi lesquelles :

- ***update*** : met à jour les dépôts de paquetages
- ***install*** <paquetage1> [<paquetage2> <paquetage3> ...] : installer un ou plusieurs paquetages
- ***remove*** <paquetage1> [<paquetage2> <paquetage3> ...] [- -purge] : supprime un ou plusieurs paquetages. Certains fichiers ne sont pas supprimés, comme par exemple les configurations de l'installation (comptes, mots de passe, ...). On aura recours à l'option ***- -purge*** pour nettoyer l'installation.
- ***autoremove*** : supprime les paquetages installés automatiquement et qui n'ont plus d'utilité, ou les éléments supprimés par l'option ***remove*** et qui sont encore présents dans le système

Gestion des services

Une fois les paquetages installés, les applications systèmes (ou services) présents dans ***/etc/init.d*** peuvent être :

- démarrés : **start**
- arrêtés : **stop**
- re-démarrés (ils sont d'abord arrêtés, puis démarrés) : **restart**
- leurs fichiers de configuration peuvent être rechargés (évite l'interruption du service pour les utilisateurs déjà connectés) : **reload**

Exemple :

```
# /etc/init.d/apache2 restart
# /etc/init.d/bind9 reload
```

On peut aussi connaître le statut (*running*, *failed*, *stopped*, etc) d'un service en utilisant l'option ***status***:

```
# /etc/init.d/mysql status
```

Beaucoup de services peuvent être gérés avec la commande ***systemctl*** (ou ***service*** pour les anciennes distributions) :

```
# systemctl restart apache2
systemctl bind9 reload
systemctl mysql status
systemctl enable apache2 //permet de l'inscrire dans les processus de démarrage de l'OS
// service apache2 restart
// service bind9 reload
// service mysql status
```

Arborescence des fichiers

Contrairement à Windows, Linux ne propose pas un ensemble de lettres de lecteur pour les différentes partitions. À la place, on trouve une arborescence unique (/) à laquelle sont rattachés les différentes partitions, lecteurs physiques (disquette, CDROM, etc) et lecteurs USB.

Pour qu'un lecteur soit visible dans cette arborescence, il faut qu'il y ait été raccroché (***monté***) par la commande ***mount***.

Le terme de monter un lecteur vient de l'époque où il fallait d'abord installer physiquement le disque avant de le mettre en route.

L'arborescence

On trouvera principalement dans l'arborescence des systèmes Linux Debian :

Lieu	Contenu
/etc	programmes et fichiers de configuration principaux
	/init.d Endroit où se trouvent les principales applications (un peu équivalent au *Program Files* de Windows)
/dev	drivers de périphériques
/media	Montage des répertoires pointant sur des unités de stockage et lecteurs disque amovibles Apparaît parfois sous le nom /mount dans d'autres distributions
/home	Stockage des répertoires de travail des utilisateurs (équivalent de *c:\Documents and settings*)
/bin	Contient les principales commandes en ligne
/var	Fichiers et données variables (données dynamiques, fichiers temporaires, pages web, etc) pour les applications
/usr	Fichiers de configuration propres à la session de l'utilisateur

Navigation dans l'arborescence

Quelques commandes pour naviguer dans les répertoires et explorer le système de fichiers :

Action	Commande	Option/remarque
Afficher le contenu d'un répertoire	**ls ** | | |::: | **ls -l ** | affiche l'utilisateur et le groupe propriétaires ainsi que les droits d'accès | |::: |ls -R**	affiche le contenu du répertoire et des sous-répertoires
	ls -all **| comme -l, y compris pour les fichiers cachés | |Créer un répertoire | mkdir <cheminRepertoire > | | |**Supprimer un répertoire** |rmdir <cheminRepertoire> | **Le répertoire doit être vide | |Supprimer un répertoire non vide** |rm -r <cheminRepertoire>| | |**Supprimer un fichier** |rm <nomFichier>**	
Se déplacer dans l'arborescence	**cd / ** | retourne à la racine | |::: |**cd .. | remonte d'un niveau dans l'arborescence | |::: |**cd <nomRepertoire> **	descend à l'intérieur de l'arborescence

Fichiers, droits d'accès, propriétaire

Commandes pour définir les autorisations et déterminer le propriétaire d'un fichier ou répertoire :

Action	Commande	Remarques
Visualiser les droits d'un élément	**ls -l**	Affiche les fichiers courants et leurs droits
	ls -all	Affiche les fichiers courants et système (cachés)
	ls -lrtR	Liste récursivement le contenu d'une arborescence en affichant les droits

Modifier le propriétaire (change owner)	**chown**	Usage :
		`chown [-R] [<nom_utilisateur>][:][<nom_groupe>] <nom_fichier>`
		Exemple : donner des droits à l'utilisateur exécutant Apache (www-data) et au groupe des superutilisateurs (root) sur le dossier Web
		`chown -R www-data:root /var/www/html`
		L'option -R réalise le travail en récursif (tous les dossiers et sous dossiers)

Modifier les droits	**chmod**	Usage :
		`chmod [-R] <nvx_droits> <nom_fichier>`
		les <nvx_droits> sont une valeur numérique de trois nombres représentant en binaire les droits (read, write, execute) pour
		* l'utilisateur,
		* son groupe
		* tous les autres.
		Par exemple :
		* 7= (111)2 : tous les droits (RWX)
		* 4 = (100)2 : droit de lecture seule (R–)
		
		on peut aussi ajouter ou supprimer un droit sans tout réécrire :
		`chmod +x -r <nomFichier>`
		L'option -R réalise le travail en récursif (tous les dossiers et sous dossiers)

Partitionner et monter un disque

Par Erwan RAMETTE et Benjamin HOHN

Partitionner un disque

Pour voir nos différents disques et partitions, on fait la commande :

```
fdisk -l
```

Pour partitionner un disque dur, il suffit d'utiliser fdisk en indiquant le chemin du disque dur en paramètre : Dans notre cas par exemple, ça sera :

```
fdisk /dev/sdb
```

Pour afficher de l'aide avec la liste des commandes disponibles, vous pouvez faire m.

Partition primaire

Une partition primaire est destiné à accueillir l'OS.

Pour créer la partition, il suffit de taper “n” (nouvelle partition), puis “p” (partition primaire) Et pour finir, de taper le numéro de la partition.

Vous pouvez ensuite, laisser les paramètres par défauts.

Partition étendue

Une partition étendue est destiné pour les datas. Une partition étendue contient des disques logiques. Les disques logiques sont quasiment identiques aux partitions, sauf que celles-ci sont définies dans le EBR (structure située en tête d’une partition logique, sur le disque dur d’un PC) au lieu de la MRB (premier secteur adressable d'un disque dur) du disque dur. L'utilisation de partitions étendues permet de créer + de 4 “partitions” par disque dur.

Pour créer une partition étendue, tapez “n” (nouvelle partition), puis “e” (partition étendue).

Vous pouvez ensuite, laisser les paramètres par défauts.

UNE FOIS PARTITIONNE, on peut formater notre partition :

```
mkfs.ext4 -b 4096 <votre partition exemple : /dev/sdd1>
```

Monter une partition

Pour monter une partition, on exécute la commande :

```
/etc/fstab
```

On va devoir rajouter une nouvelle ligne de ce type dans le fichier : UUID=..... var/data ext4 defaults 0 2

Voici l'exemple :



Utilisation des fichiers de configuration

Contrairement à Windows où les configurations des services se passent par interface graphique et sont stockées dans la base de registres, Linux utilise encore des fichiers texte. Voici quelques éléments pour comprendre leur fonctionnement et utiliser au mieux une approche modulaire.

Contenu/Syntaxe

Un fichier de configuration est éditable à partir d'un utilitaire en mode texte (vi, vim, nano, pico ou autre, voir annexe pour quelques commandes) ou un éditeur plus complet si on dispose d'une interface graphique (par exemple *gedit* sous Debian/Ubuntu).

Les éléments de syntaxe suivants sont très courants :

Elément	Usage
#	Commentaire textuel qui donne des explications sur les options
;	Commente une ligne de paramétrage. A dé commenter si on veut activer l'option
<intitulé> ... </intitulé>	Encadre une description complexe (même principe que pour les balises HTML ou la structure d'enregistrement en algorithmique). On sépare différentes valeurs par des espaces

Approche modulaire

Pour éviter d'encombrer un fichier de configuration, on a recours à une **externalisation** (comme on le fait pour le CSS qui sort la mise en page du HTML). On crée donc des fichiers externes qui ne prennent en compte qu'une partie de la configuration, et on y fait appel dans un fichier central grâce à la clause ***include***.

Exemple pour apache2.conf

```
# inclue les fichiers de configuration contenus dans le sous-dossier conf.d
include conf.d/
# inclue les déclarations d'hôtes virtuels contenus dans le sous-dossier sites-enabled
include sites-enabled/
# Inclue la liste des ports d'écoute contenue dans le fichier ports.conf
include ports.conf
```

Exemple pour DNS

Le fichier de base du service Bind est ***named.conf*** dont le contenu est :

```
include "/etc/bind/named.conf.options"; #options de configuration du serveur
include "/etc/bind/named.conf.local"; #zones gérées par le serveur
include "/etc/bind/named.conf.default-zones"; #zones par défaut (localhost,
résolution inverse, etc)
```

Nettoyage des fichiers et Versionning

Pour rendre la lecture des fichiers de configuration aisée, on ***épure*** leur contenu en supprimant toute option non utilisée, tout commentaire inutile De manière à pouvoir revenir à la situation initiale, on a coutume de faire une copie d'un fichier de configuration avant de le modifier. On utilise la commande ***cp*** pour copier un fichier vers un double avec un nouveau nom.

Exemple

```
cp    /etc/apache2/apache2.conf    /etc/apache2/apache2.old1
```

Editeur VI

Pour manipuler les fichiers systèmes, qui sont des fichiers texte, il est nécessaire de recourir à un éditeur de texte. Même quand aucune interface graphique n'est disponible, un outil rudimentaire bien qu'offrant d'innombrables possibilités permet toute manipulation sur un fichier texte : l'éditeur vi.

Le mode commande (auquel on revient en appuyant Echap) propose diverses actions dont les plus utiles sont présentées ci-dessous :

action	commande	remarques
insérer du texte	i	Passe l'éditeur en mode insertion. Taper à nouveau i (ou Echap) pour repasser en mode commande attention : les flèches sont désactivées pour les déplacements
supprimer une ligne	dd	Ces commandes ne marchent pas si on est en mode insertion
supprimer le caractère sous le curseur	x	
supprimer le caractère à gauche du curseur	X	
enregistrer les modifications	:w	
quitter vi	:q	
enregistrer et quitter	:wq	
quitter sans enregistrer	:q!	Il n'y a pas d'invite à confirmer
Rechercher du texte vers le bas du fichier	/texte_cherché	On tapera à nouveau / pour chercher la prochaine occurrence
Rechercher du texte en arrière	?texte_cherché	On tapera à nouveau ? pour chercher l'occurrence précédente

Des versions un peu plus riches sont disponibles, qui proposent des menus accessibles à partir du clavier et parfois de la colorisation syntaxique : *vim*, *nano*, *pico*.

GNU-Nano

GNU Nano est un éditeur de texte, pas un traitement de texte !

Voir plus sur https://openclassrooms.com/fr/courses/43538-reprenez-le-controle-a-laide-de-linux/39267-nano-lediteur-de-texte-du-debutant

Pour éxécuter GNU Nano, il suffit de taper « nano » dans la console linux.

Voici les raccourcis les plus importants :
Ctrl + G : afficher l'aide ;
Ctrl + K : couper la ligne de texte (et la mettre dans le presse-papier) ;
Ctrl + U : coller la ligne de texte que vous venez de couper ;
Ctrl + C : afficher à quel endroit du fichier votre curseur est positionné (numéro de ligne...) ;
Ctrl + W : rechercher dans le fichier ;
Ctrl + O : enregistrer le fichier (écrire) ;
Ctrl + X : quitter Nano.

Pour installer le correcteur :

```
apt install spell aspell-fr
```

Pour installer plus de possibilité pour le correcteur orthographique (Cela permet d'ajouter des proposition de correction) :

```
nano --speller="aspell -c"
```

Scripting sous Linux

Manipulations de fichiers / dossiers

Action	Commande
Copier un fichier	cp
Déplacer un fichier ou le renommer	mv
Supprimer un fichier	rm

Parcours de fichier

Le parcours de fichier consiste à définir les informations à lire sur une ligne et à effectuer ensuite un traitement sur les variables ainsi nommées.

```
#! /bin/bash
echo off
while [IFS=<séparateur>] read champ1 champ2
  do
    echo "Ceci est le champ n°1: " $champ1
    echo "Et voici le champ 2 : " $champ2
done< [</chemin>]<fichierAlire>
```

- **#! /bin/bash** : pour afficher la syntaxe en couleur
- **echo off** : pour ne pas afficher les commandes qui s'exécutent au fur et à mesure
- **while … read champ1 champ2** : boucle de parcours qui précise la structure du contenu de chaque ligne
- **IFS=<séparateur>** : option qui permet de définir le séparateur des champs dans un fichier structuré (CSV). Le séparateur est généralement un des caractères : , ;. L'espace est le séparateur par défaut
- **Do** : début du code à exécuter dans la boucle
- **$champ1** : variable contenant la première information de la ligne
- **echo -e** : *echo* réalise un affichage écran, -e permet d'ajouter un retour à la ligne
- **done** : fin de la boucle
- ** <** : indique qu'on prend le contenu de ce qui suit
- **[</chemin>]<fichierAlire>** : fichier dont on exploite le contenu (avec chemin éventuel)

Gestion des dates

On peut gérer l'affichage des dates avec la commande ***date*** : La syntaxe est diffrente selon les distributions (tester avec ou sans les “”):

```
date +"<format>"
#ou
date +<format>
```

Le format utilise les paramètres suivants :

- %d : affiche le jour
- %m : affiche le mois
- %y : affiche l'année sur deux caractères (18 pour 2018)
- %Y : affiche l'année sur quatre caractères (2018)
- %H : affiche l'heure
- %M : affiche les minutes
- %S : affiche les secondes

Exemples :

```
date +"%d/%m/%y"
```

Affichera 13/02/18.

```
date +"%Y-%m-%d"
```

Affichera 2018-02-13

Variables

Les variables dans les scripts se déclarent par leur nom, et s'utilisent avec le préfixe $.

Exemple :

```
#il n'y a pas d'espace entre la variable et le caractère "="
mavariable="bonjour"
echo $mavariable
```

La concaténation de chaine dans une variable s'écrit par la simple juxtaposition des portions de texte :

Exemple :

```
laDate=$(date +%Y%m%d)
nomFichier="fichier"$laDate".csv"
```

From:
https://wiki.sio.bts/ - **WIKI SIO : DEPUIS 2017**

Permanent link:
https://wiki.sio.bts/doku.php?id=linux&rev=1617091214

Last update: **2021/03/30 08:00**