Marc Brayer

Ligne de commande

Contents

Utilisation de GNU/Linux

Nous allons débuter par l'étape d'utilisation du système GNU/Linux , je vais me concentrer sur l'interface ligne de commande. La raison est simple l'interface graphique est plus ou moins intuitif , mais surtout la ligne de commande peut être utilisé peut importe le type d'interface graphique que vous utilisez (Gnome , KDE , Xfce , …) . L'autre raison très importante si vous désirez travailler sur un serveur en tant normal l'interface graphique n'est pas installé pour réduire l'emprunte mémoire , car non ou peu utilisé , ceci fait en sorte que la mise à jour du système est plus rapide , …

Donc débutons par comprendre la ligne de commande , je vais utiliser l'interpréteur de commande **bash** , le plus répandu sur les distributions GNU/Linux , je vous laisse le plaisir de découvrir Zsh et autre interpréteur cool :P.

Apprivoiser la ligne de commande

Une interface en ligne de commandes peut servir aussi bien pour lancer l'exécution de divers logiciels au moyen d'un interpréteur de commandes, que pour les dialogues avec l'utilisateur de ces logiciels. C'est l'interaction fondamentale entre un homme et un ordinateur (ou tout autre équipement informatique).

Lorsqu'une interface est prête à recevoir une commande, elle l'indique par une invite de commande. Celle-ci, parfois désignée par l'anglicisme prompt, consiste en quelques caractères, en début de ligne (généralement, le nom de compte de l'utilisateur, et/ou l'unité logique par défaut, et/ou le chemin par défaut, et/ou date, …), se terminant par un caractère bien connu (souvent «] », « # », « $ » ou « > »), invitant l'utilisateur à taper une commande.

Commentaire : vous pouvez utiliser la ligne de commande en tout temps, personnellement quand je démarre mon ordinateur j'ai TOUJOURS un terminal qui démarre. Si je veux démarrer Firefox , ou vlc je le fait en ligne de commande , ma femme trouve toujours ça drôle car je le fait même si l'explorateur de fichiers fut déjà ouvert … Par habitude je ne clique pas sur les icônes :D.

Invite de commande (prompt)

Si vous démarrez un terminal, vous aurez tout de suite l'invite de commande. Beaucoup de distribution présente l'invite comme suit :

```
UserName@Hostname:Répertoire_courant$
```

Si vous êtes connecté avec l'utilisateur root, ce devrait ressemblé à :

```
root@Hostname_de_la_machine:Répertoire_courant#
```

- **UserName** : représente évidement le nom de l'utilisateur en cours d'utilisation
- **Hostname** : Le nom de la machine qui attend les instructions à la ligne de commande.
- **Répertoire_courant** : Le répertoire où vous vous trouvez. Si vous avez le caractère ~ , ceci équivaut au répertoire personnelle de l'utilisateur. Donc si l'utilisateur ce nomme thomas, ~ == /home/thomas
- **$ ou #** : $ indique que vous êtes connecté comme un utilisateur "régulier". # indique que vous êtes connecté avec le compte administrateur.

Évoluer dans l'arborescence et manipulation de fichier

Comme disait si bien Mr. Miyagi dans Karaté Kid :

> "First learn stand, then learn fly. Nature rule Daniel son, not mine" - Mr. Miyagi

Donc avant de lancer des commandes dans tous les sens voyons la base :P

Afficher le répertoire courant (pwd)

Pour savoir dans quelle répertoire vous vous trouvez, vous pouvez utiliser la commande pwd (print working directory) :

```
$ pwd
/home/Bob
```

Car moi aussi parfois je me perd dans le système :P.

Changer de répertoire (cd)

La commande cd (change directory) vous permet de changer de répertoire ,

```
$ cd .                    # . désigne le répertoire courant
$ cd ..                   # .. désigne le répertoire parent
$ cd /                    # / désigne le répertoire racine
 ------------------------------------------------------------------------------
$ cd /tmp                 # désigne le répertoire tmp appartenant à la racine
$ cd tmp                  # désigne le répertoire tmp du répertoire courant
$ cd ../tmp               # désigne le répertoire tmp du répertoire parent du
   répertoire courant
 ------------------------------------------------------------------------------
$ cd ~                    # permet de revenir dans son répertoire de travail
   (home directory)
$ cd                      # idem
```

Liste les fichiers d'un répertoire (ls)

la commande **ls** (list) liste le contenu du répertoire. Un grand nombre d'options existent pour cette commande, afin de faciliter l'affichage du résultat. Par défaut, **ls** liste le contenu du répertoire courant, mais il est aussi possible de lui fournir le répertoire ou fichier à lister.

```
 # Liste le contenu du répertoire courant /home/alex
utilisateur@hotname:/home/alex$  ls
Desktop        Downloads  Music     Public      Videos
Documents      fichiers   Pictures  Templates

 # Liste le contenu du répertoire /tmp
utilisateur@hotname:/home/alex$  ls /tmp
pulse-2L9K88eMlGn7  tmpksCRJ5   ssh-BkXkFipk2242  fichiers_temporaire
un_autre_exemple
```

Option intéressante :

- -l (long) : liste détaillée des fichiers. Affiche les permissions , le propriétaire , la taille et la date du fichier

```
utilisateur@hostname:/home/alex $ ls -l /etc/
total 1636
drwxr-xr-x  3 root root      4096 Mar 12  2013 acpi
-rw-r--r--  1 root root      2981 Apr 25  2011 adduser.conf
drwxr-xr-x  2 root root     32768 Nov  7 16:06 alternatives
-rw-r--r--  1 root root       395 Jun 20  2010 anacrontab
drwxr-xr-x  7 root root      4096 Jul 26 16:01 apache2
-rw-r--r--  1 root root       112 Jun 22  2007 apg.conf
drwxr-xr-x  6 root root      4096 Apr 25  2011 apm
drwxr-xr-x  3 root root      4096 Sep 20 08:19 apparmor
drwxr-xr-x  8 root root      4096 Nov  5 09:47 apparmor.d
drwxr-xr-x  5 root root      4096 Nov  5 09:46 apport
drwxr-xr-x  6 root root      4096 Oct 23 13:03 apt
-rw-r-----  1 root daemon     144 Jun 27  2010 at.deny
drwxr-xr-x  2 root root      4096 Aug 24  2012 at-spi2
drwxr-xr-x  3 root root      4096 Aug 24  2012 avahi
-rw-r--r--  1 root root      9085 Jun 12  2012 avserver.conf
-rw-r--r--  1 root root      2076 Apr  3  2012 bash.bashrc
-rw-r--r--  1 root root     58753 Oct  4  2011 bash_completion
drwxr-xr-x  3 root root     16384 Nov  7 16:06 bash_completion.d
drwxr-sr-x  2 root bind      4096 Aug  7 16:35 bind
....... [texte tronqué] .....
```

- -a (all) : liste tous les fichiers, y compris les fichiers "cachés". Un fichier "caché" est un fichier qui débute par un "."

```
utilisateur@hotname:/home/alex$  ls -a
Desktop        .bashrc       .bash_profile    Downloads  Music     Public
   Videos
Documents    fichiers       Pictures          Templates   .app      .gnupg
```

- –color= auto / none : Permet d'afficher ou non les couleurs avec ls , la majorité des distributions active la coloration par défaut

Option Combinées , il est possible de combiner des options :

- **-l -t -r** ou **-ltr** : Affiche le résultat en format long (**-l**) , Tri le résultat par la date de modification du fichier contrairement au nom par défaut (**-t**) le fichier le plus récent s'affichera en premier le plus vieux en dernier, inverse le résultat (**-r**) ceci afin d'avoir le fichier le plus récent affiché en dernier.

```
  # sans le reverse
utilisateur@hotname:/home/alex$  ls -lt /etc
...... [texte tronqué] ...
drwxr-xr-x  3 root root      4096 May  9  2010 insserv
-rw-r--r--  1 root root       645 Mar  7  2010 ts.conf
-rw-r--r--  1 root root     15752 Jul 25  2009 ltrace.conf
-rw-r--r--  1 root root       112 Jun 22  2007 apg.conf
-rw-r--r--  1 root root     10852 Apr 27  2007 gnome-vfs-mime-magic
-rw-r--r--  1 root root      1343 Jan  9  2007 wodim.conf
-rw-r--r--  1 root root      2064 Nov 23  2006 netscsid.conf
utilisateur@hotname:/home/alex$

   # avec :)
 utilisateur@hotname:/home/alex$  ls -ltr /etc
 ...... [texte tronqué] ...
 -rw-------  1 root root      1822 Nov 29 15:51 shadow-
 -rw-------  1 root root      1164 Nov 29 15:51 group-
 -rw-------  1 root root       959 Nov 29 15:51 gshadow-
 -rw-r--r--  1 root root      1176 Nov 29 15:51 group
 -rw-------  1 root root      2650 Nov 29 15:51 passwd-
 -rw-r-----  1 root shadow     967 Nov 29 15:51 gshadow
 -rw-r-----  1 root shadow    1944 Nov 29 15:51 shadow
 -rw-r--r--  1 root root      2653 Nov 29 15:51 passw
```

- **-l** –**block-size=M** ou **-h** : Affiche le résultat en format long (**-l**) et fournie la taille des fichiers en Megs , beaucoup plus facile à lire

```
utilisateur@hostname:~$ ls -l -h
drwxr-xr-x  6 xerus xerus 4.0K Jul 22 21:19 gits
-rw-r--r--  1 xerus xerus 1.7K Mar 23  2013 GNUstepDefaults
-rw-r--r--  1 xerus xerus  33M Sep 17 12:31 go.tar.gz
-rw-rw-r--  1 xerus xerus  87K Nov 21 12:53 hand_spacewalk.png
```

Afficher le contenu d'un fichier

Afficher tout le fichier (cat)

La commande **cat** (concatenate) affiche le contenu d'un fichier ou de plusieurs fichiers concaténés à l'écran.

```
utilisateur@hostname:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
```

```
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
....... [texte tronqué] .....
statd:x:114:65534::/var/lib/nfs:/bin/false
postfix:x:115:124::/var/spool/postfix:/bin/false
utilisateur@hostname:~$
```

Option intéressante :

- **-n** ou –**number** (number) : permet d'afficher le numéro de ligne

```
utilisateur@hostname:~$ cat -n  /etc/passwd
     1    root:x:0:0:root:/root:/bin/bash
     2    daemon:x:1:1:daemon:/usr/sbin:/bin/sh
     3    bin:x:2:2:bin:/bin:/bin/sh
     4    sys:x:3:3:sys:/dev:/bin/sh
     5    sync:x:4:65534:sync:/bin:/bin/sync
....... [texte tronqué] .....
    35    statd:x:114:65534::/var/lib/nfs:/bin/false
    36    postfix:x:115:124::/var/spool/postfix:/bin/false
```

Afficher le fichier avec un paginer (less / more)

Le problème de **cat** est qu'il peut être compliqué de visualiser un fichier s'il est très grand. En effet, **cat** fera dérouler le terminal du début à la fin ; il peut donc être difficile à lire. Pour solutionner ce problème, nous avons **less** et more qui s'offrent à nous . Personnellement, j'utilise beaucoup plus **less** que more, car le premier à plus d'options que le deuxième :P .

```
 # utilisation de less
utilisateur@hotname:~$ less /var/log/dmesg

 # utilisation de less , affiche les numéros de ligne
utilisateur@hotname:~$ less -N /var/log/dmesg

 # utilisation de more
utilisateur@hotname:~$ more /var/log/dmesg
```

Afficher le début et la fin d'un fichier (head / tail)

Nous ne désirons pas toujours avoir l'intégralité d'un fichier. Par moment, afficher un nombre X de ligne au début ou à la fin d'un fichier peu être suffisant. La commande **head** et **tail**, avec l'option -n (number), nous permet d'avoir ce comportement . Par défaut, si le nombre de ligne n'est pas spécifié, le système affiche 10 lignes.

```
 # HEAD
utilisateur@hostname:~$ head -n 4 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
```

```
 # TAIL
utilisateur@hostname:~$ tail  /etc/passwd
rtkit:x:106:116:RealtimeKit,,,:/proc:/bin/false
sshd:x:107:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:108:117:MySQL Server,,,:/nonexistent:/bin/false
minidlna:x:109:119:MiniDLNA server,,,:/var/lib/minidlna:/usr/sbin/nologin
ntp:x:110:120::/home/ntp:/bin/false
fetchmail:x:111:65534::/var/lib/fetchmail:/bin/false
tester:x:1001:1001:,,,:/home/tester:/bin/bash
avahi:x:112:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
postfix:x:115:124::/var/spool/postfix:/bin/false
```

Option intéressante de **tail**:

- **-f** ou –**follow** : permet d'afficher à l'écran les lignes qui se rajoutent au fichier de manière continue. Ceci est TRÈS intéressant, principalement pour la visualisation des logs.

Création de répertoire (mkdir)

La création d'un répertoire peut être réalisé en utilisant la commande **mkdir** (make directory), en donnant le nom du répertoire à créer en argument, comme suit.

```
 # Notation absolu
utilisateur@hostname:/home/utilisateur$ mkdir /tmp/Nom_du_repertoire

 # Notation relative
utilisateur@hostname:/home/utilisateur$ mkdir Nom_du_repertoire
utilisateur@hostname:/home/utilisateur$ mkdir ../../tmp/Nom_du_repertoire
```

Option intéressante :

- **-p** : permet de créer une arborescence de répertoire plus rapidement ; mkdir, par défaut, ne crée qu'un répertoire à la fois. Par exemple, si je suis dans le répertoire /home/alex et que je désirer créer la hiérarchie de répertoire /home/alex/images/2013/06/18, voici comment faire :

```
 # Commande en ERREUR
utilisateur@hotname:/home/alex$ mkdir images/2013/06/18
mkdir: cannot create directory 'images/2013/06/18': No such file or directory

 #solution
utilisateur@hotname:/home/alex$ mkdir -p images/2013/06/18
```

Copie de fichier ou répertoire (cp)

La commande **cp** (copie) nous permet de réaliser la copie d'un ou plusieurs fichiers ou répertoires.

```
 # Copie du fichier passwd dans /tmp
utilisateur@hotname:/home/alex$ cp /etc/passwd /tmp

 # Copie du fichier passwd et changement de nom du fichier pour copie_pass
utilisateur@hotname:/home/alex$ cp /etc/passwd /tmp/copie_pass
```

```
 # Copie de plusieurs fichier dans un répertoire, quand l'on copie plusieurs
     fichier la destination doit obligatoirement être un répertoire
utilisateur@hotname:/home/alex$ cp /etc/passwd  /etc/fstab /etc/group /tmp/

 # Copie d'un répertoire avec l'option -r récursive
utilisateur@hotname:/home/alex$ cp /etc/network /tmp
```

Options intéressantes :

- **-r** (récursive) : telle que présenté plus tôt, nous permet de faire une copie récursivement des fichiers
- **-l** ou –**link** (link) : ne réalise pas la copie du fichier à proprement parlé, mais créer un Hard Link du fichier vers l'inode correspondant sur le disque.
- –**parents** : cette option nous permet d'indiquer à la commande **cp** de recréer la structure de répertoire lors de la copie ; économisant l'utilisation de la commande de création de répertoire **mkdir**

```
utilisateur@hotname:/home/alex$ ls -R /tmp/testing
utilisateur@hotname:/home/alex$ cp --parent /usr/share/debianutils/shells
   /usr/share/gimp/2.0/images/gimp-splash.png /etc/network/interfaces
   /tmp/testing/
utilisateur@hotname:/home/alex$ ls -R /tmp/testing
/tmp/testing/:
etc usr

/tmp/testing/etc:
network

/tmp/testing/etc/network:
interfaces

/tmp/testing/usr:
share

/tmp/testing/usr/share:
debianutils gimp

/tmp/testing/usr/share/debianutils:
shells

/tmp/testing/usr/share/gimp:
2.0

/tmp/testing/usr/share/gimp/2.0:
images

/tmp/testing/usr/share/gimp/2.0/images:
gimp-splash.png
```

Suppression de fichier ou répertoire (rm)

La commande **rm** (remove) permet la suppression de fichier ou répertoire

```
 # suppression du fichier photo1.png
utilisateur@hotname:/home/alex$ rm Lefichier

 # pour la suppression de répertoire il faut ajouter l'option -r pour
    récursive
 # ici la suppression est réalisé sur le répertoire photos_noel
utilisateur@hostname:/home/alex$ rm -r photos_noel
```

Déplacement de fichier ou renommage (mv)

La commande **mv** (move) permet de déplacer les fichiers d'un répertoire à l'autre ou de renommé le fichier

```
 # Déplace un fichier d'un répertoire à l'autre
utilisateur@hotname:/home/alex$  mv /home/alex/Documents/billetterie.png
   Projets/GrosProblem/

 # renommer un fichier
utilisateur@hotname:/home/alex$  mv  Projets/GrosProblem/billetterie.png
   Projets/GrosProblem/loterie.png
```

Éditeur minimaliste (pico/nano)

En attendant de présenter un "vrai" éditeur de texte du moins l'un des plus puissants disponible sur GNU/Linux et Unix c'est à dire **VI** vous pouvez utiliser pico ou nano les deux font appelle au même binaire lors de l'installation sous Ubuntu. nano vous offre l'équivalent de notepad , un éditeur de texte simple pour modifier des fichiers voici à quoi il ressemble :

```
utilisateur@hostname:~$ nano /etc/passwd
....... [texte tronqué] .....
sshd:x:107:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:108:117:MySQL Server,,,:/nonexistent:/bin/false
minidlna:x:109:119:MiniDLNA server,,,:/var/lib/minidlna:/usr/sbin/nologin
ntp:x:110:120::/home/ntp:/bin/false
fetchmail:x:111:65534::/var/lib/fetchmail:/bin/false
tester:x:1001:1001:,,,:/home/tester:/bin/bash
avahi:x:112:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
postfix:x:115:124::/var/spool/postfix:/bin/false




  [ Read 36 lines (Warning: No write permission) ]
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur
   Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To
   Spell
```

Exécution de commandes sous l'identité de l'administrateur (sudo / su)

Vous avez peut-être voulu éditer un fichier dans le répertoire /etc/, pour y modifier un paramètre, ou désiré expérimenter des commandes trouvées sur internet. Malheureusement, tous furent des échecs, car le système indique un problème de permission.

Deux méthodes existent pour exécuter des commandes sous l'utilisateur administrateur (**root**) ; il y a la bonne et la mauvaise !!! Commençons par la bonne ; si vous venez d'installer votre système **ubuntu**, votre premier utilisateur a le droit d'utiliser la commande **sudo** (**S**with**U**ser**DO**). En plus d'avoir les permissions d'exécuter cette commande, il peut tout faire avec cette dernière ; **Sudo** permet d'attribuer des droits de manière granulaire à un utilisateur. Nous verrons en détails cet aspect dans la section administration du système .

Pour visualiser les permissions que nous avons :

```
 # affiche les permissions de l'utilisateur Alex.
alex@hostname:~$ sudo -l
[sudo] password for alex:
Matching Defaults entries for alex on this host:
    env_reset,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User alex may run the following commands on this host:
    (ALL : ALL) ALL
 # Alex dans la situation peut executer n'importer quelle commande sur le
    système sous l'utilisateur root ou autre
```

Donc, si nous désirons modifier un fichier dans le répertoire /etc/ avec la commande nano, nous utiliserons la commande suivante :

```
alex@hostname:~$ sudo nano /etc/Le_fichier
```

C'est une bonne pratique d'utiliser la commande sudo, car :

- l'ensemble des commandes sont cataloguées ; il est donc possible de savoir quelle commande fut exécutée , à quelle moment et par qui
- permet de limiter les commandes disponibles pour une personne, afin de protéger l'utilisateur et le système d'une erreur.

L'autre méthode est de Switché d'utilisateur, de passer complètement en root. Le problème avec ceci est que nous perdons toute trace des opérations réalisées et qu'il n'y a plus de filé si un erreur de manipulation est réalisée. C'est l'équivalent à se connecter comme Administrateur sur une machine plutôt que d'utiliser la fonction de "RunAs" sous Microsoft Windows. Bien entendu, dans certaine situation, nous n'avons pas le choix. C'est parfois le cas lors de l'installation de logiciels en dehors des packages de la distribution.

Voici comment faire le MAL :P, de passer sous l'utilisateur root:

```
alex@hotname:~$ su -
    # ou
alex@hotname:~$ su
```

Localiser un fichier

Bon voilà, vous êtes content, votre système fonctionne bien. Sauf que vous voulez aller plus loin ou vous avez un problème ; vous recherchez donc sur internet ou demandez à un ami comment réaliser une opération. Ce

dernier, content de vous répondre, rétorque simplement quelque chose du genre : " Ha, c'est simple! Genre, change le paramêtre Xy dans le fichier smb.conf pour la valeur true" . Bon super vous notez le tout et, arrivé devant la machine, vous constatez que vous ne savez pas où se trouve le fichier smb.conf. Trop habitué, votre collègue a oublié de fournir le chemin absolue. Voici comment trouver votre fichier :

Recherche dans l'arborescence (find)

La commande **find**, comme son nom, l'indique permet de rechercher dans le système un fichier. Bien entendu, une recherche peut être longue s'il y a beaucoup de fichiers à traiter. **find** nous permet de rechercher selon plusieurs critères : le nom du fichier, sa date de modification, son propriétaire, ses permissions, sa taille, etc.

La structure de la commande est : **find** Répertoire_de_recherche Pattern. Si aucun répertoire de recherche n'est fourni, le système commence la recherche depuis le répertoire courant .

Si nous reprenons mon exemple mentionné plus tôt, nous aurions la commande suivante :

```
    # comme le fichier est un .conf, je fais une recherche dans le répertoire
       /etc contenant les fichiers de configuration du système
utilisateur@hostname:~$ find /etc/ -name "smb.conf"

    # Si je ne trouve pas je vais faire une recherche plus général , à partir
       de la racine.
utilisateur@hostname:~$ find / -name "smb.conf"

    # Si je ne trouve toujours pas , je vais élargir ma recherche , en
       présument que le fichier commence par smb et fini par .conf
utilisateur@hostname:~$ find / -name "smb*.conf"

    # voici un exemple du résultat dans la vie Réelle
utilisateur@hostname:~$ find /etc -name "smb.conf"
find: `/etc/lvm/backup': Permission denied
find: `/etc/lvm/cache': Permission denied
find: `/etc/rsnapshot/keys': Permission denied
find: `/etc/xdg/menus/applications-merged': Permission denied
find: `/etc/openvpn/ytriaV2': Permission denied
find: `/etc/ssl/private': Permission denied
/etc/samba/smb.conf
```

La dernière commande retourne des erreurs, car je ne suis pas administrateur du système. Cependant, à la fin, j'ai trouvé le fichier, qui se trouve dans /etc/samba .

Voici un tableau avec les principales options

Option	Signification
-name	Recherche par nom de fichier.
-type	Recherche par type de fichier.
-user	Recherche par propriétaire.
-group	Recherche par appartenance à un groupe.
-size	Recherche par taille de fichier.
-atime	Recherche par date de dernier accès.
-mtime	Recherche par date de dernière modification.
-ctime	Recherche par date de création.
-perm	Recherche par autorisations d'accès.

Recherche avec la base de donnée updatedb (locate)

!!! Attention, ceci n'est pas disponible sur toutes les distributions !!! Bon, **find** c'est bien, mais si j'ai un répertoire avec énormément de fichiers, la recherche peut être très longue… Heureusement, plusieurs distributions installent le système d'**updatedb** par défaut. Celui-ci exécute toutes les nuits la commande **find** à partir du **Root** (/) et stocke l'information dans une base de données, pour usage ultérieur.

Donc, si nous reprenons la recherche du fichier smb.conf, ceci donne :

```
utilisateur@hostname:~$ locate smb.conf
/etc/samba/smb.conf
/usr/share/man/man5/smb.conf.5.gz
/usr/share/samba/smb.conf
/var/lib/ucf/cache/:etc:samba:smb.conf

utilisateur@hostname:~$
```

Comme vous pouvez le constater, le résultat fournit l'information pour le répertoire **/etc**, mais aussi pour **/usr** et **/var**. Ceci dans un temps record! Attention, **locate** a quelques limitations :

- Aucun résultat ne sera affiché pour les fichiers créés depuis la dernière mise à jour de la base de données.
- Recherche uniquement par nom de fichier.
- La mise à jour de la BD augmente le load de la machine.
- Le résultat peut être erroné si le système n'a pas mis à jour sa BD. Par exemple, si le système été éteint pendant la planification de synchronisation.

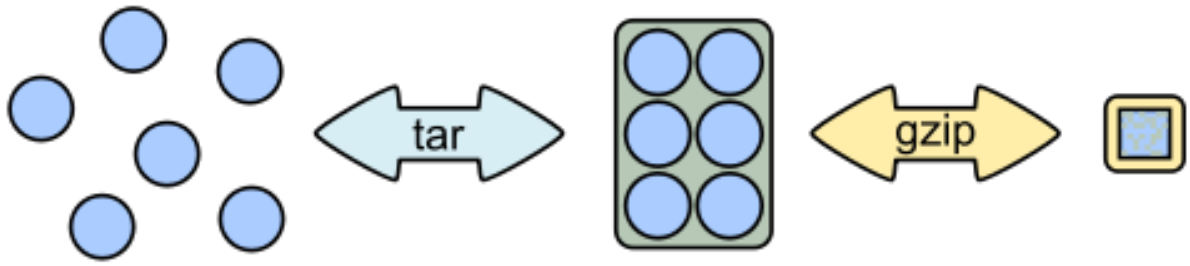
Archivage et extraction de données (tar / gzip / bz2 / zip)

Le système d'archivage de prédilection sous Unix est **tar** (**T**ape **AR**chiver) pour l'archivage.

Un fichier d'archive créé par **tar** n'est pas compressé. On appelle parfois le fichier d'archivage créé un **tarball**. **Tar** préserve les droits, le propriétaire et le groupe des fichiers et des répertoires. Il permet également de sauvegarder les liens symboliques et les fichiers spéciaux orientés bloc ou caractère.

Pour l'essentiel, le format employé consiste en une concaténation du contenu des fichiers. Chaque fichier est précédé d'un en-tête de 512 octets ; cette taille correspondant à la taille d'un bloc dans la version 7 du système de fichiers Unix.

Pour améliorer l'efficacité de l'écriture sur bande magnétique, les blocs de 512 octets sont groupés par 20 par défaut, produisant des blocs de 10 Ko. Le bloc final est complété par des zéros binaires.



Souvent, un fichier créé par **tar** est ensuite compressé par un outil de compression de données. Les formats les plus courants sont :

Mode compression	Extention unix	extention Dos
compress	.tar.Z	.taz

Mode compression	Extention unix	extention Dos
gzip	.tar.gz	.tgz
bzip2	.tar.bz2	.tbz
lzma	.tar.lz	.tlz
lzma2	.tar.xz	.txz
7zip	.tar.7z	

```
 # Réalisation d'une archive :
$ tar -cvf fichier.tar MonRepertoire1 [MonRepertoire2... ]

 # Afficher le contenu d'une archive
$ tar -tf archive.tar

 # Extraire les fichiers d'une archive
$ tar -xvf archive.tar

 # Il est possible d'extraire uniquement des fichiers d'une archive
$ tar -xvf archive.tar fichier1 fichier 2

  ############ Tar avec compression ##############

 # Réalisation d'une archive compressé avec gzip
$ tar -zcvf fichier.tar.gz MonRepertoire1 [MonRepertoire2... ]

 # Afficher l'archive compressé avec gzip
$ tar -ztvf fichier.tar.gz

 # Extraire l'archive compressé avec gzip
$ tar -zxvf fichier.tar.gz

 # Réalisation d'une archive compressé avec bzip2
$ tar -jcvf fichier.tar.gz2 MonRepertoire1 [MonRepertoire2... ]

 # Afficher l'archive compressé avec gzip
$ tar -jtvf fichier.tar.gz2

 # Extraire l'archive compressé avec gzip
$ tar -jxvf fichier.tar.bz2
```

Changement de mot de passe

La commande passwd modifie le mot de passe du compte des utilisateurs. Un utilisateur normal peut uniquement modifier le mot de passe de son compte (le plus souvent). Le Super-utilisateur a le pouvoir de changer le mot de passe de chaque compte. passwd permet aussi d'associer une période de validité aux comptes et mots de passe, et de les modifier. En plus de permettre le changement de mot de passe, passwd permet de verrouiller un compte, de définir une période d'expiration …

```
  # l'utilisateur peut changer son propre password
utilisateur@hostname:~$ passwd
enter new UNIX password:
```

```
retype new UNIX password:
passwd: password updated successfully

 # L'administrateur (root) peut manipuler les comptes des autres utilisateurs

    # ici changement du mot de passe de thomas
root@hostname:~# passwd thomas

   # verrouillage du compte de joe
root@hostname:~# passwd -l joe
```

Options intéressantes:

- **-l**, –**lock** : verrouille le compte spécifié. Cette option désactive complètement le mot de passe en le préfixant d'un ´!´ (aucune valeur cryptée selon le mécanisme utilisé par **passwd** ne peut donc y correspondre).
- **-u**, –**unlock** : déverrouille le mot de passe du compte indiqué. Cette option réactive le mot de passe en le positionnant à la valeur qui existait avant l'utilisation de l'option **-l**. **-w**,
- **-n**, –**mindays** MIN_JOURS : définit le nombre minimum de jours entre chaque changement de mot de passe à MIN_JOURS. Une valeur égale à zéro dans ce champ indique que l'utilisateur peut changer son mot de passe lorsqu'il le souhaite.
- –**warndays** DUREE_AVERTISSEMENT : fixe le nombre de jours avant que le changement de mot de passe ne soit obligatoire. DUREE_AVERTISSEMENT est le nombre de jours précédant la fin de la validité du mot de passe et durant lesquels l'utilisateur sera averti que son mot de passe est sur le point d'arriver en fin de validité.