

Linux: Guide Utilisateur

Prise en main de la ligne de commande Bash

I. Commandes sur l'arborescence du système de fichiers

Astuce : Pour toute information sur une commande, tapez `$ man commande` où « commande » correspond à la commande souhaitée. Si le manuel n'existe pas pour la commande, tapez `$ commande help --all` ou `$ commande -help`.

`$ ls`

Liste les répertoires et fichiers du répertoire courant (celui dans lequel on se trouve).

```
pierre@robot: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
pierre@robot:~$ ls  
bootstrap-3.3.7-dist  khabzaoui_crypt.pdf  postfix.txt  
Bureau               Modèles              Public  
cinema               mum_mail_site.txt    raw_iso  
corebook.pdf         Musique              sources_code  
Documents            os_tree              Téléchargements  
droit                ovh.log              TEST  
GLMF                 ownCloud             Vidéos  
guide_linux          passwd_raspi.txt     VirtualBox VMs  
hack.txt             perl                 VNC-6.0.2-Linux-x64  
Images              PhpstormProjects  
pierre@robot:~$
```

`$ ls -l`

Liste les répertoires et fichiers avec les informations sur les permissions (rights, size, owner, owner group, modification date, etc.).

```
pierre@robot: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
pierre@robot:~$ ls -l  
total 4224  
drwxr-xr-x 13 pierre pierre 4096 avril 21 19:20 bootstrap-3.3.7-dist  
drwxrwxrwx  7 root  root  4096 avril 19 13:28 Bureau  
drwxr-xr-x  2 pierre pierre 4096 mars 30 00:58 cinema  
drwxr-xr-x  2 pierre pierre 4096 mai 18 21:54 corebook.pdf  
drwxrwxrwx  2 root  root  4096 avril  3 20:58 Documents  
-rw-r--r--  1 pierre pierre 1045931 mai  3 12:59 droit  
drwxr-xr-x  2 pierre pierre 4096 mai 18 21:55 GLMF  
drwxr-xr-x  2 pierre pierre 4096 mai 18 21:54 guide_linux  
-rw-r--r--  1 pierre pierre 115 mai 19 11:00 hack.txt  
-rw-r--r--  1 pierre pierre  87 avril 28 17:55 Images  
drwxr-xr-x  2 pierre pierre 4096 mai 19 11:03 khabzaoui_crypt.pdf  
-rw-r--r--  1 pierre pierre 3135508 mai  6 17:37 Modèles  
drwxr-xr-x  2 pierre pierre 4096 avril 12 15:07 postfix
```

`$ ls -a`



Liste également les fichiers cachés (se reconnaissant par le « . »).

```
pierre@robota: ~  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
pierre@robota:~$ ls -a  
.  
..  
.bash_history  
.bash_logout  
.bashrc  
bootstrap-3.3.7-dist  
bormux  
Bureau  
.cache  
cinema  
communa  
.config  
corebook.pdf  
Documents  
droit  
hack.txt  
ICEauthority  
Images  
.java  
khabzaoui_crypt.pdf  
.local  
lock_guide_linux#  
Modèles  
.mozilla  
mum_mail_site.txt  
Musique  
.nano  
os_tree  
ovh.log  
ownCloud  
.profile  
Public  
.purple  
raw_iso  
scilab  
sources_code  
.ssh  
Téléchargements  
TEST  
.thumbnails  
Vidéos  
.vim  
.viminfo  
VirtualBox VMs  
.vnc
```

Note : **-a** et **-l** pour la commande **ls** par exemple sont des **OPTIONS** précisées dans leur forme raccourcie. Elle peuvent être précisée autrement par exemple **--all** pour **-a**. Il est conseillé de se renseigner sur les différentes options des commandes par le biais des commandes **man** ou **help**.

Astuce : On peut également préciser les options sous une forme contractée par exemple \$ ls -all/-a -l peut s'écrire \$ ls -al.

PATH :

Chemin absolu : Un chemin absolu désigne le chemin parcouru dans l'arborescence depuis la racine.

Exemple : Je suis dans le répertoire /home/pierre. C'est un chemin absolu. Si je désire aller dans le répertoire Test/ je tape: **cd /home/pierre/Test**

Chemin relatif : Un chemin relatif désigne le chemin parcouru depuis l'arborescence du répertoire courant.

Exemple : Je suis dans le répertoire /home/pierre. Si je désire aller dans le répertoire Test/ je tape: **\$ cd Test/**.

\$ pwd

```
pierre@robota:~$ pwd  
/home/pierre  
pierre@robota:~$ █
```

Indique le positionnement du répertoire courant dans l'arborescence globale (chemin absolu).

\$ cd



Traduction : « change directory ». Permet de changer de répertoire.

```
pierre@robota:~$ cd /home/pierre/PhpstormProjects/  
pierre@robota:~/PhpstormProjects$
```

\$ cd ~

```
pierre@robota:/$ cd ~  
pierre@robota:~$ ls  
bootstrap-3.3.7-dist  Images          postfix  
bormux               khabzaoui_crypt.pdf postfix.txt  
Bureau               Modèles        Public  
cinema               mum_mail_site.txt raw_iso  
Communs              Musique         scripts
```

Permet de revenir au répertoire utilisateur, équivalent à
\$ cd /home/nom_utilisateur.

\$ cd /

```
pierre@robota:~$ cd /  
pierre@robota:/$ ls  
bin    etc      initrd.img.old  lost+found  opt    run    sys    var  
boot   home     lib             media       proc   sbin   tmp    vmlinuz  
dev    initrd.img lib64           mnt         root    srv    usr    vmlinuz.old  
pierre@robota:/$
```

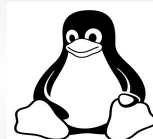
Permet de revenir à la racine.

\$ cd ..

Permet de revenir un revenir au répertoire précédent dans arborescence.
Peut s'utiliser avec des récurrences.

Exemple : \$ cd ../../..

```
pierre@robota:~/PhpstormProjects$ cd /  
pierre@robota:/$ cd /home/pierre/PhpstormProjects/first/.idea/scopes/  
pierre@robota:~/PhpstormProjects/first/.idea/scopes$ cd ../../..  
pierre@robota:~/PhpstormProjects$ ls  
first  
pierre@robota:~/PhpstormProjects$ pwd  
/home/pierre/PhpstormProjects  
pierre@robota:~/PhpstormProjects$
```



II. Manipulation des données dans le système de fichier

1.1 Créer des dossiers et des fichiers, manipulation des chaînes de caractères :

\$ mkdir nom_répertoire

```
pierre@robota:~/Test$ mkdir test
pierre@robota:~/Test$ ls
test
pierre@robota:~/Test$
```

Permet de créer un répertoire.

\$ touch nom_fichier.extension

Permet de créer un fichier vide et de spécifier son extension.

```
pierre@robota:~/Test$ touch test.txt
pierre@robota:~/Test$ ls
test test.txt
pierre@robota:~/Test$
```

\$ echo « on écrit quelque chose »

Le résultat de la commande echo affiche le contenu de la parenthèse dans la console.

\$ echo « test » > test.txt

Le contenu de la commande echo est ajouté au fichier. Si le fichier existe il est écrasé. Pour l'ajouter à la suite d'un fichier déjà existant taper :

\$ echo « test » >> test.txt

```
pierre@robota:~$ echo quelque chose
quelque chose
pierre@robota:~$ echo "test" > test.txt
pierre@robota:~$ cat test.txt
test
pierre@robota:~$ echo "autre test" >> test.txt
pierre@robota:~$ cat test.txt
test
autre test
```

1.2 La concaténation :

\$ cat nom_fichier

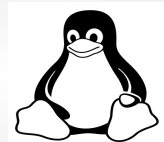
Affiche le contenu d'un fichier en console. Pour l'afficher page par page, taper :

\$ cat nom_fichier | less

Note : la commande **\$ cat**, bien qu'elle serve également à afficher le contenu des fichiers est avant tout un opérateur de concaténation (d'où son nom).

\$ cat fic.ext fic2.ext > fic3.ext

```
pierre@robota:~/Test$ echo "ceci est un test de concaténation" > test2.txt
pierre@robota:~/Test$ cat test.txt test2.txt > test3.txt
pierre@robota:~/Test$ cat test
test/      test2.txt  test3.txt  test.txt
pierre@robota:~/Test$ cat test
test/      test2.txt  test3.txt  test.txt
pierre@robota:~/Test$ cat test3.txt
test
ceci est un autre test
ceci est un test de concaténation
pierre@robota:~/Test$ █
```



1.3 Déplacement, copie, suppression et exécution des données :

\$ cp fic /rep/new_fic

Copie un fichier d'un emplacement du disque à un autre. Ne pas spécifier de nouveau nom de fichier si l'on souhaite garder le même.

```
pierre@robota:~/test$ cp test.txt nouveau_test
pierre@robota:~/Test$ ls
nouveau_test  test  test2.txt  test3.txt  test.txt
pierre@robota:~/Test$ cp test.txt testnouveau_test
test/          test2.txt  test3.txt  test.txt
pierre@robota:~/Test$ cp test.txt test/nouveau_test
```

\$ rm fichier

Supprime un fichier.

Utiliser l'option **-R** (pour « récursif ») si l'on souhaite supprimer un répertoire et son contenu.

\$ rm -R[r] /répertoire

```
pierre@robota:~/Test$ rm test.txt
pierre@robota:~/Test$ ls
nouveau_test  test  test2.txt  test3.txt
pierre@robota:~/Test$ rm -R test/
pierre@robota:~/Test$ ls
nouveau_test  test2.txt  test3.txt
pierre@robota:~/Test$ █
```

Utiliser l'option **-f** ou **--force** pour effectuer une suppression si un message d'erreur est rencontrée.

\$ mv fichier /rep/new_name_fichier

```
pierre@robota:~/Test$ mv fic Testo/ceci_est_un_fichier_déplacé_et_renommé
pierre@robota:~/Test$ mv test3.txt
test3.txt  Testo/
pierre@robota:~/Test$ mv test3.txt
test3.txt  Testo/
pierre@robota:~/Test$ mv test3.txt ceci_est_un_fichier_renommé
pierre@robota:~/Test$ ls
ceci_est_un_fichier_renommé  Testo
pierre@robota:~/Test$ ls Testo/
ceci_est_un_fichier_déplacé_et_renommé  nouveau_test
pierre@robota:~/Test$ █
```

Permet de déplacer un fichier et donne également la possibilité de le renommer.

mv pour « move » est d'ailleurs une des rares commandes qui permet de renommer un fichier sans en créer un nouveau. Si l'on est intéressé par les possibilités de renommage on se focalisera davantage sur **rename**.

De la même façon que pour **cp** on pourra utiliser les options **-f** et **-r** pour des actions forcées ou récursives.

\$./executable

Si je désire exécuter un script ou un programme je me positionne dans le répertoire de celui-ci et l'instancie par « **./nom.[bin/ext]** ».

```
pierre@robota:~/VNC-6.0.2-Linux-x64$ ./vncviewer
pierre@robota:~/VNC-6.0.2-Linux-x64$ █
```

Il est parfois utile de supprimer les données d'un fichier sans pour autant supprimer celui-ci.

On pourra, à cet usage, utiliser :

rm fichier && touch file

Le **&&** que vous découvrez ici signifie : « Si ma première commande s'est bien effectuée, exécute la seconde ». Cette ligne aura donc pour effet de supprimer un fichier pour le recréer aussitôt.

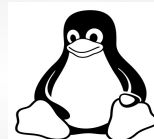
Cependant, nous pouvons être plus subtils :

/dev/null est un fichier « trou noir ». Tout ce qui est affecté à **/dev/null** ou inversement disparaît. C'est un fichier spécial.

Aussi, la commande :

\$ cat /dev/null > file

aura pour effet de supprimer les données du fichier cible sans l'effacer. En fait, la redirection « **>** » apparente dans la syntaxe réécrit « **file** » avec le contenu de **/dev/null**, soit rien.



1.4 Faire des liens :

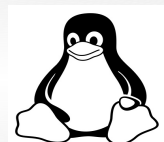
\$ ln

Permet de faire des liens : physiques ou symboliques.

Lien physique : permet de donner plusieurs noms/chemin d'accès, à un même fichier en pointant sur un numéro de fichier, (en interne Linux enregistre les fichiers sur la base d'un numéro et pas sur la base d'un nom). Un fichier peut donc avoir plusieurs noms, et existera tant qu'il a au moins un nom. Contrairement aux liens symboliques, ils ne peuvent pointer que vers un autre élément du même système de fichiers.

Lien symbolique : permet d'attribuer un autre chemin d'accès à un fichier en pointant sur un nom de fichier

\$ ln



1.5 Connaître son architecture :

\$ uname -a

Indique quelle est l'image du **kernel** chargée sur le systèmes

\$ uname -m

Indique seulement le type d'architecture utilisée par le système.

```
pierre@robota:~$ uname -a
Linux robota 4.9.0-3-amd64 #1 SMP Debian 4.9.25-1 (2017-05-02) x86_64
GNU/Linux
pierre@robota:~$ uname -m
x86_64
pierre@robota:~$ █
```



2. La commande mount :

Sur un **rootfilesystem** linux, pour accéder à des ressources disques extérieures à la partition système ou à des périphériques (USB, cdrom, etc.) il est nécessaire de « monter » les partitions de ces périphériques sur un « point de montage » qui sera accessible depuis la partition courante.

Ainsi, on « monte un système de fichier ».

Traditionnellement les périphériques physiques que l'on aura à « monter » apparaissent dans **/dev**.

```
pierre@robota:~$ ls /dev/
autofs          network_latency tty1    tty4    ttyS3
block           network_throughput tty10   tty40   uhid
bsg             null            tty11   tty41   uinput
btrfs-control  nvram           tty12   tty42   urandom
bus             port            tty13   tty43   v4l
char            ppp             tty14   tty44   vboxdrv
console         psaux           tty15   tty45   vboxdrvu
core            ptmx            tty16   tty46   vboxnetctl
cpu             pts             tty17   tty47   vboxusb
cpu_dma_latency random           tty18   tty48   vcs
cuse            rfkill          tty19   tty49   vcs1
disk            rtc             tty2    tty5    vcs2
dri             rtc0            tty20   tty50   vcs3
fb0             sda             tty21   tty51   vcs4
fd             sda1            tty22   tty52   vcs5
full            sda2            tty23   tty53   vcs6
fuse            sda3            tty24   tty54   vcsa
hpet            sda4            tty25   tty55   vcsa1
hugepages       sda5            tty26   tty56   vcsa2
initctl         sda6            tty27   tty57   vcsa3
input           sda7            tty28   tty58   vcsa4
kmsg            sda8            tty29   tty59   vcsa5
kvm             sda9            tty3    tty6    vcsa6
```

La plupart des ressources apparaissant sont déjà montées et utilisées par le système mais ce n'est pas le cas de toutes.

Pour exemple : les périphériques **sdxX** (ou X est une lettre quelconque déterminant le périphérique et X le numéro de la partition) sont ici **sda1**, **sda2**, etc. les partitions du **hdd** du poste sur lequel nous effectuons les tests. Si une clé USB devait être connectée elle apparaîtrait ici en **/dev/sdb**.

Supposons que l'on souhaite accéder à **/dev/sda7** qui nous sert de partition de stockage et ne contient aucun système :

```
# mount /dev/sa4 /mnt/point_de_montage
```

Conventionnellement le répertoire système `/mnt/` sert de point de montage pour les ressources externes.

```
root@robota:~# mount /dev/sda7 /mnt/corea/
root@robota:~# ls /mnt/corea/
core/  corea/  coreo/  corex/
root@robota:~# ls /mnt/corea/
backup.tar.bz2  Images      lost+found  shell
COPY_MAIN      ISO&RAW     movies      Visual_Basic&Python_Excel
html_          LEARN       pierre
```

Nous accédons désormais au contenu de `/dev/sda7` par le répertoire `/mnt/corea` sur lequel la ressource a été montée.

Pour « démonter » le périphérique :

```
# umount /dev/nom_du_peripherique
```

```
root@robota:~# umount /dev/sda7
```

Note : Si la ressource montée est définie par son **UUID**, Utilisez **lsblk -o +UUID,PARTUUID** pour vérifier que les **UUID** sont vraiment uniques sur le système.

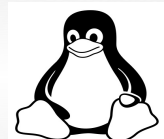
```
# mount -a [-t type] [-O liste_options]
```

Monte tous les systèmes de fichiers indiqués dans **fstab** (ou uniquement ceux du type indiqué et possédant ou non les bonnes options), sauf ceux dont la ligne contient l'option **noauto**. En ajoutant l'option **-F**, **fork** sera invoqué par **mount** pour que tous les systèmes de fichiers soient montés simultanément.

Lors du montage d'un système de fichiers mentionné dans **fstab** ou **mtab**, il suffit d'indiquer soit le point de montage, soit le périphérique en ligne de commande.

Source :

```
man mount
```

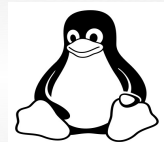


2.1 Quelques options :

- **rw** : monte le système de fichiers en lecture/écriture
- **sync** : toutes les entrées et sorties du système de fichier doivent être réalisées de façon synchrone.
- **remount** : tenter de remonter un système de fichier déjà monté.
- **exec** : permettre l'exécution de fichiers binaires
- **strictatime** : demander explicitement une mise à jour complète des horodatages d'accès.
- **owner** : autoriser un utilisateur a monter le système de fichier si celui-ci est propriétaire du périphérique.

Exemple d'options combinées :

```
# mount -o remount,rw /dev/test /mnt/rep_test
```



3. La commande `mkfs` :

`mkfs` est une commande permettant de créer un système de fichier linux.

Elle est particulièrement utile pour reformaté un périphérique en tant que système de fichier voulu telle une clé usb par exemple.

Aujourd'hui jugée obsolète ce chapitre traitera de `mkfs.type`.

3.1 `mke2fs` :

Crée un système de fichier `ext2/ext3/ext4`.

Nous ne traiterons pas des différences entre ces types de système de fichier dans cet ouvrage. Sachez néanmoins que les rapports disques sont plus courts en `ext2` qu'en `ext3` ou `ext4` et qu'`ext2` ne prend pas en charge la **journalisation**.

Sachez aussi qu'aujourd'hui, `ext4` le système de fichier le plus utilisé sous Linux. C'est aussi le plus récent et il est compatible avec la majorité des supports de stockage.

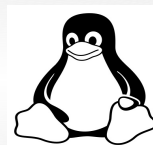
Ces **rootfilesystems** ne sont lisibles que pour un OS Linux ou « based on Linux ».

Dans cet exemple on va formater une clé usb située sur `/dev/sdb` :

```
# mkfs.ext4 -c -F -U UUID -L nom_nouveau_volume /dev/sdb
```

Ici on va créer un système de fichier `ext4` sur `dev/sdb/`, vérifier les blocs et clusters défectueux (`-c`), forcer la création si une erreur devait être retournée (fréquent si le périphérique est non-vidé par exemple) avec l'option `-F`, lui attribuer un **UUID** et un nom (**Label**).

4. La commande dd :





5. Badblocks :

Cet utilitaire analyse et liste les secteurs défectueux ou sains du disque, les clusters, les blocs.

```
$ badblocks -v /dev/sda4 > bad_blocks_file
```

```
pierre@robota:~$ sudo badblocks /dev/sda4 > badblocks_file
pierre@robota:~$ find /home/pierre/ -iname "badblocks_file"
/home/pierre/badblocks_file
```

Analyse en mode verbeux la partition **/dev/sda4** du disque et écrit le diagnostic dans «bad_blocks_file».

On lit ensuite le diagnostic écrit dans le fichier à l'aide de la commande :

```
$ fsck -t ext4 -l bad_blocks_file /dev/sda4
```

où ext4 correspondra au système de fichier en vigueur sur le périphérique.

```
pierre@robota:~$ sudo fsck.ext4 -l bad_blocks_file /dev/sda4
e2fsck 1.43.4 (31-Jan-2017)
/dev/sda4 est monté.
e2fsck: Ne peut continuer, arrêt immédiat.
```

On observe dans l'exemple ci-dessus que si le périphérique à analyser est monté, **fsck** nous retourne une erreur et ne peut s'exécuter proprement.

6. Fscck :

fsck est utilisé pour vérifier et éventuellement réparer un ou plusieurs systèmes de fichiers Linux. Le système_de_fichiers peut être un nom de périphérique (par exemple **/dev/hdc1**, **/dev/sdb2**), un point de montage (par exemple **/**, **/usr**, **/home**), une étiquette (« **label** ») **ext2** ou un identifiant **UUID** (par exemple **UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd** ou **LABEL=root**). Le programme **fsck** essayera de fonctionner en parallèle pour les systèmes de fichiers situés sur des disques physiques différents afin de minimiser la durée totale de vérification.

Si aucun système de fichiers n'est précisé sur la ligne de commande et que l'option **-A** n'est pas indiquée, **fsck** vérifiera les systèmes de fichiers présents dans **/etc/fstab**. C'est équivalent à préciser les options **-As**.

Source :

man fsck

Se référer au manuel pour les codes de retour.

7. Conversion de fichiers :

7.1 Convertir des fichiers DOS/UNIX :

Pour convertir un fichier DOS Windows en un fichier Linux, on utilisera la commande **sed**.

```
$ sed 's/^M$//' fichier_dos > fichier_unix
```

L'exécution de cette commande crée un nouveau fichier lisible par Linux sur la base du fichier source après conversion par **sed**.

Pour faire «**^M**» dans la console, faire **Ctrl+V** puis **Ctrl+M**.

Pour convertir un fichier UNIX en fichier DOS :

```
$ sed 's/$'\`echo \\r`/' unix_file > dos_file
```

Conversion en terminal **bash**.

```
$ sed «s/$'\`echo -e \\r`/' unix_file > dos_file
```

Conversion en terminal **ksh**.

```
$ sed 's/$'\`r`/' unix_file > dos_file
```

Conversion en **gsed** 3.02.80 ou supérieur.

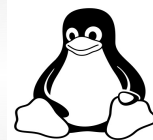
Des utilitaires de conversion tel que **dos2unix** et **unix2dos** sont également disponibles sur certaines distributions.

```
$ dos2unix DOSfile UNIXfile
```

Conversion de DOS à UNIX.

```
$ unix2dos UNIXfile DOSfile
```

Conversion de UNIX à DOS.



8. Suppression définitive et sûre des données :

On peut supprimer des données de manière « relativement » sûre à l'aide de la commande **shred**.

Cet utilitaire est obtenu lors de l'installation du paquet **coreutils**.

apt-get install coreutils

shred écrase de façon sécurisée les données d'un fichier par l'algorithme de Gutmann.

Pour un « shredding » rapide :

\$ shred -vzn 0 /directory/fichier_à_écraser

La commande va « **overwrite** » (trad. « écraser avec les données ») le fichier en effectuant une seule itération.

```
pierre@robota:~$ shred -vzn 0 shred_test_file
shred: shred_test_file : passe 1/1 (000000)...
pierre@robota:~$ cat shred_test_file
pierre@robota:~$
```

Le fichier existe toujours mais les données ont été intégralement effacées.

\$ shred -vzn 3 /directory/fichier_à_écraser

Le processus de shredding va être répété 3 (itérations par défaut de **shred**) fois en utilisant les nombres aléatoires (« **random numbers** »).

L'option **-u** provoquera la suppression du fichier après écrasement et réécriture des données. L'option **-z** écrira des « 0 » par-dessus les anciennes données écrasées. Sachez toutefois que l'usage de cette option prendra à **shred** 4 fois plus de temps pour effectuer l'opération.

On peut effectuer un nombre **n** d'itérations souhaitées.

\$ shred --remove -vzn 5 file_to_shred

Voir capture plus bas.

--remove est le nom complet de l'option abrégée **-u**.

On notera que pour utiliser la contraction des options, celles-ci doivent s'effectuer dans un ordre précis. En effet la commande s'exécute linéairement et sa syntaxe rencontre une erreur si **-n** ne termine pas les



options puisqu'un nombre définies d'itérations doivent être précisées.

Exemples :

```
pierre@robota:~$ shred -vznu 5 shred_test_file
shred: invalid number of passes: « u »
pierre@robota:~$
```

```
pierre@robota:~$ shred --remove -vzn 5 shred_test_file
shred: shred_test_file : passe 1/6 (random)...
shred: shred_test_file : passe 2/6 (000000)...
shred: shred_test_file : passe 3/6 (random)...
shred: shred_test_file : passe 4/6 (ffffff)...
shred: shred_test_file : passe 5/6 (random)...
shred: shred_test_file : passe 6/6 (000000)...
shred: shred_test_file : suppression
shred: shred_test_file : renommé en 0000000000000000
shred: 0000000000000000 : renommé en 0000000000000000
shred: 0000000000000000 : renommé en 0000000000000000
shred: 0000000000000000 : renommé en 0000000000000000
shred: 0000000000000000 : renommé en 0000000000000000
shred: 000000000000 : renommé en 000000000000
shred: 000000000000 : renommé en 0000000000
shred: 0000000000 : renommé en 00000000
shred: 00000000 : renommé en 000000
shred: 000000 : renommé en 00000
shred: 00000 : renommé en 0000
shred: 0000 : renommé en 000
shred: 000 : renommé en 00
shred: 00 : renommé en 0
shred: shred_test file : supprimé
pierre@robota:~$
```

Les données du fichier ont donc été 5 écrasées, réécrites avec des nombres aléatoires et des 0, puis celui-ci a été supprimé.

En cas de données sensibles, ne pas hésiter à faire appel à **shred** lors de la suppression.

III. Gestionnaire de paquets APT

APT (Advanced Packaging Tool) est une collection d'outils permettant de gérer les logiciels installés sur une machine de façon relativement simple et complète. C'est un système de gestion de paquet robuste et élégant, qui a longtemps fait la fierté de Debian.

Le fonctionnement est le suivant : **APT** conserve une liste des paquets installés, avec leur version, et leur état. Par exemple, lorsque vous installez un paquet, celui-ci est marqué comme *manuellement installé*. En revanche, si ce paquet dépendait d'une bibliothèque, alors celle-ci a été installée automatiquement, et marquée comme telle.

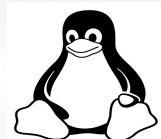
En parallèle, APT conserve une liste des paquets installables. Il récupère cette liste depuis les dépôts précisés dans les fichiers **/etc/apt/sources.list** et **/etc/apt/sources.list.d/*** (ces dépôts sont la plupart du temps des sites internet, mais peuvent également être des cdrom ou des miroirs locaux).

Installation depuis les sources :

Depuis des fichiers **.deb** téléchargés manuellement (bien que cette dernière méthode soit déconseillée.)

Source :

<https://debian-facile.org/doc:systeme:apt>



apt-get update

La commande **update** permet de resynchroniser un fichier d'index répertoriant les paquets disponibles et sa source.

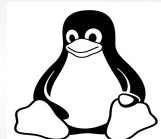
```
pierre@robota:/$ sudo apt-get update
Réception de:1 http://ftp.fr.debian.org/debian stretch InRelease [186
kB]
Réception de:2 http://security.debian.org/debian-security stretch/upd
ates InRelease [62,9 kB]
Atteint:3 http://packages.prosody.im/debian stable InRelease
Réception de:4 http://ftp.fr.debian.org/debian stretch/main Sources.d
iff/Index [27,9 kB]
Réception de:5 http://ftp.fr.debian.org/debian stretch/main i386 Pack
ages.diff/Index [27,9 kB]
Réception de:6 http://ftp.fr.debian.org/debian stretch/main amd64 Pac
kages.diff/Index [27,9 kB]
Réception de:7 http://ftp.fr.debian.org/debian stretch/main Translati
on-fr.diff/Index [27,8 kB]
Réception de:8 http://ftp.fr.debian.org/debian stretch/main Sources 2
017-05-19-0229.06.pdiff [1 256 B]
Réception de:8 http://ftp.fr.debian.org/debian stretch/main Sources 2
017-05-19-0229.06.pdiff [1 256 B]
Réception de:9 http://ftp.fr.debian.org/debian stretch/main i386 Pack
ages 2017-05-19-0229.06.pdiff [7 759 B]
Réception de:9 http://ftp.fr.debian.org/debian stretch/main i386 Pack
ages 2017-05-19-0229.06.pdiff [7 759 B]
```

apt-get upgrade

La commande **upgrade** permet d'installer les mises à jour disponibles de tous les paquets présents sur le système en utilisant les sources énumérées dans **sources.list**.

apt-get dist-upgrade

```
pierre@robota:~$ sudo apt-get dist-upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les NOUVEAUX paquets suivants seront installés :
  linux-headers-4.9.0-3-amd64 linux-headers-4.9.0-3-common
  linux-image-4.9.0-3-amd64 xserver-xorg-legacy
Les paquets suivants seront mis à jour :
  at-spi2-core bash bind9-host binutils cpp-6 cryptsetup
  cryptsetup-bin deluge deluge-common deluge-gtk
  dh-strip-nondeterminism dnsutils dpkg dpkg-dev fonts-opensymbol
```

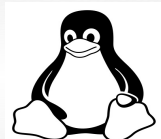


La commande **dist-upgrade** effectue la fonction **upgrade** en y ajoutant une gestion intelligente des changements de dépendances dans les nouvelles versions des paquets. Le gestionnaire de paquets **apt-get** possède un système « intelligent » de résolution des conflits et il essaye, quand c'est nécessaire, de mettre à niveau les paquets les plus importants avant les paquets les moins importants.

```
pierre@robota:/$ sudo apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
  linux-headers-amd64 linux-image-amd64 xorg xserver-xorg
Les paquets suivants seront mis à jour :
  at-spi2-core binutils cryptsetup cryptsetup-bin
  dh-strip-nondeterminism gcr gdm3 gir1.2-atspi-2.0 gir1.2-gck-1
  gir1.2-gcr-3 gir1.2-gdm-1.0 gir1.2-javascriptcoregtk-4.0
  gir1.2-mate-panel gir1.2-webkit2-4.0 hplip hplip-data iso-codes
  libatk-adaptor libatk-bridge2.0-0 libatspi2.0-0 libcryptsetup4
  libfile-stripnondeterminism-perl libgck-1-0 libgcr-3-common
  libgcr-base-3-1 libgcr-ui-3-1 libgdm1 libhpmud0
  libjavascriptcoregtk-4.0-18 libmate-panel-applet-4-1
  libnss-myhostname libpam-systemd libpq5 libsane-hpaio libsbcl
  libservlet3.1-java libsystemd0 libsystemd0:i386 libudev1
  libudev1:i386 libwebkit2gtk-4.0-37 libwebkit2gtk-4.0-37-gtk2
  libzmq5 linux-compiler-gcc-6-x86 linux-kbuild-4.9 linux-libc-dev
  mate-icon-theme mate-panel mate-panel-common mate-settings-daemon
  mate-settings-daemon-common printer-driver-hpcups
  printer-driver-hpijs printer-driver-postscript-hp python-cairo
  ssl-cert systemd systemd-sysv udev unattended-upgrades x11-common
  xserver-xorg-input-all xserver-xorg-video-all
63 mis à jour, 0 nouvellement installés, 0 à enlever et 4 non mis à j
our.
Il est nécessaire de prendre 128 ko/62,8 Mo dans les archives.
Après cette opération, 8 581 ko d'espace disque supplémentaires seron
t utilisés.
Souhaitez-vous continuer ? [0/n] █
```

apt-get install paquet

Permet d'installer un paquet depuis les dépôts renseignés dans le `/etc/apt/sources.list`. La tabulation complétera la typologie du paquet ou renseignera sur ses différences en cas d'un trop grand nombre de paquets à la typologie similaire.



```

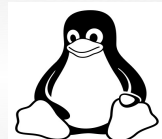
pierre@robota:/$ sudo apt-get install ruby
[sudo] Mot de passe de pierre :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
The following additional packages will be installed:
  fonts-lato libruby2.3 rake ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.3
  rubygems-integration
Paquets suggérés :
  ri ruby-dev bundler
Les NOUVEAUX paquets suivants seront installés :
  fonts-lato libruby2.3 rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.3
  rubygems-integration
0 mis à jour, 11 nouvellement installés, 0 à enlever et 66 non mis à
jour.
Il est nécessaire de prendre 6 192 ko dans les archives.
Après cette opération, 27,1 Mo d'espace disque supplémentaires seront
utilisés.
Souhaitez-vous continuer ? [0/n] █

```

apt-get build-dep paquet

Cette commande installera, si elle les trouve, toutes les dépendances (paquets nécessaires au bon fonctionnement) du paquet spécifié en opérande.

«**apt-get build-dep** installe ou supprime des paquets dans le but de satisfaire les dépendances de construction d'un paquet source. Par défaut, les dépendances sont satisfaites pour la construction native du paquet. Au besoin, une architecture hôte peut être indiquée avec l'option **--host-architecture**.»




```

root@robota:~# apt-get build-dep rails
Lecture des listes de paquets... Fait
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
 devscripts gem2deb gem2deb-test-runner libgmp-dev libgmpxx4ldbl
 libmariadbclient18 libruby2.3 rake ruby ruby-actionpack
 ruby-actionview ruby-activemodel ruby-activerecord
 ruby-activesupport ruby-all-dev ruby-arel ruby-atomic ruby-bcrypt
 ruby-blankslate ruby-builder ruby-bundler ruby-concurrent
 ruby-daemons ruby-dalli ruby-delayed-job ruby-did-you-mean
 ruby-erubis ruby-eventmachine ruby-globalid ruby-hike ruby-i18n
 ruby-json ruby-loofah ruby-mail ruby-metaclass ruby-mime-types
 ruby-minitest ruby-mocha ruby-molinillo ruby-multi-json
 ruby-mysql2 ruby-net-http-persistent ruby-net-telnet
 ruby-nokogiri ruby-pkg-config ruby-polyglot ruby-power-assert
 ruby-rack ruby-rack-cache ruby-rack-test
 ruby-rails-deprecated-sanitizer ruby-rails-dom-testing
 ruby-rails-html-sanitizer ruby-setup ruby-sprockets
 ruby-sprockets-rails ruby-sqlite3 ruby-test-unit ruby-thor
 ruby-thread-safe ruby-tilt ruby-treetop ruby-tzinfo ruby2.3
 ruby2.3-dev rubygems-integration sqlite3
0 mis à jour, 67 nouvellement installés, 0 à enlever et 67 non mis à
jour.
Il est nécessaire de prendre 10,7 Mo/10,7 Mo dans les archives.
Après cette opération, 47,9 Mo d'espace disque supplémentaires seront
utilisés.
Souhaitez-vous continuer ? [0/n] █

```

apt-get remove --purge paquet

Supprime un paquet ainsi que ses sources. Les fichiers de configuration sont également effacés.

```

pierre@robota:/$ sudo apt-get remove --purge rubygems-integration
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets suivants seront ENLEVÉS :
 rubygems-integration*
0 mis à jour, 0 nouvellement installés, 1 à enlever et 66 non mis à j
our.
Après cette opération, 20,5 ko d'espace disque seront libérés.
Souhaitez-vous continuer ? [0/n] 0█

```

apt-get source paquet

Apt récupère les paquets sources et télécharge ensuite dans le répertoire courant la version la plus récente du paquet si l'option **-t** est spécifiée.

```
root@robota:~# apt-get source rsync
Lecture des listes de paquets... Fait
Nécessité de prendre 921 ko dans les sources.
Réception de:1 http://ftp.fr.debian.org/debian stretch/main rsync 3.1
.2-1 (dsc) [1 676 B]
Réception de:2 http://ftp.fr.debian.org/debian stretch/main rsync 3.1
.2-1 (tar) [893 kB]
Réception de:3 http://ftp.fr.debian.org/debian stretch/main rsync 3.1
.2-1 (diff) [26,4 kB]
921 ko réceptionnés en 11s (78,4 ko/s)
dpkg-source: info: extraction de rsync dans rsync-3.1.2
dpkg-source: info: extraction de rsync_3.1.2.orig.tar.gz
dpkg-source: info: extraction de rsync_3.1.2-1.debian.tar.xz
```

Si l'option **--compile** est spécifiée, le paquet est compilé en un binaire **.deb** avec **dpkg-buildpackage** pour l'architecture définie par l'option **--host-architecture**. Si **--download-only** est spécifié, le paquet source n'est pas décompacté.

apt-get check

Sert à **apt** pour diagnostiquer les dépendances défectueuses et mettre à jour le cache des paquets.

apt-get clean

La commande **clean** nettoie le référentiel local des paquets récupérés. Elle supprime tout, excepté le fichier de verrou situé dans **/var/cache/apt/archives/** et **/var/cache/apt/archives/partial/**.

apt-get autoclean

Supprime uniquement les paquets qui ne peuvent plus être téléchargés et qui sont inutiles.

apt-get autoremove

Supprime les paquets installés devenus obsolètes ou inutiles dans le but de satisfaire les dépendances d'autres paquets et qui ne sont plus nécessaires.



```

pierre@robota:~$ sudo apt-get autoremove
[sudo] Mot de passe de pierre :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets suivants seront ENLEVÉS :
  linux-image-4.8.0-2-amd64
0 mis à jour, 0 nouvellement installés, 1 à enlever et 0 non mis à jo
ur.
Après cette opération, 187 Mo d'espace disque seront libérés.
Souhaitez-vous continuer ? [O/n] █

```

Options :

De nombreuses options existent pour les commandes **apt**. Pour une documentation plus exhaustive que dans cet ouvrage on conseillera de se référer au manuel d'apt, récemment traduit en français.

Quelques unes parmi les plus notables :

--arch-only :

Ne traiter que les dépendances de construction indépendantes de l'architecture.

--reinstall :

Réinstaller les paquets déjà installés dans leur configuration la plus récente.

-d, --download-only :

Les paquets sont récupérés mais ne sont ni dépaquetés ni installés.

-f, --fix-broken :

Demande de réparer un système où existent des dépendances défectueuses.

-y, --yes, --assume-yes :

Répondre « oui » automatiquement aux questions et exécuter apt de manière non interactive.

-b, --compile, --build :

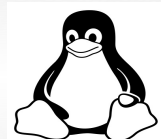
Cette commande compile un paquet source après l'avoir récupéré.

--force-yes :

Forcer l'acceptation.

Source :

man apt, écrit par Jason Gunthorpe



V. L'administration du réseau

Préambule :

L'utilisation des outils venant du paquet **net-tools** (**ifconfig**, **arp**, **miitools**, etc.) est devenue obsolète et leur utilisation est même dépréciée sur les distributions GNU/Linux récentes.

Dorénavant, depuis un certains temps c'est le paquet **iproute2** qui est installé par défaut pour la gestion du réseau sous GNU/Linux, notamment avec l'aide de la commande **ip**.

Source :

<https://memo-linux.com/ip-la-commande-linux-pour-gerer-son-interface-reseau/> posted by fred.

\$ ip addr add 192.168.1.2/24 dev wlp2s0

Permet d'attribuer une adresse IPV4 sur l'interface cible.

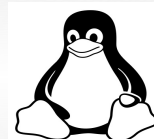
```
pierre@robota:~$ ip addr add 192.168.1.2/24 dev wlp2s0
RTNETLINK answers: Operation not permitted
pierre@robota:~$ sudo ip addr add 192.168.1.2/24 dev wlp2s0
pierre@robota:~$ ip -4 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    inet 192.168.43.244/24 brd 192.168.43.255 scope global dynamic wlp2s0
        valid_lft 3390sec preferred_lft 3390sec
    inet 192.168.1.2/24 scope global wlp2s0
        valid_lft forever preferred_lft forever
```

\$ ip -4 addr show

Permet de visualiser les propriétés réseau IPV4 de l'interface cible ou de toutes les interfaces réseau.

On peut utiliser l'option **-4** pour ne retenir que les adresses IPV4 et **-o** pour n'afficher le résultat que sur une ligne. L'option **-c** sert à la coloration syntaxique.

Note : sur les distributions Linux récentes, l'interface wireless wlp2s0 remplace l'interface wlan0.



\$ ip addr del 192.168.1.2/24 dev wlp2s0

```
pierre@robota:~$ sudo ip addr add 192.168.1.2/24 dev wlp2s0
pierre@robota:~$ sudo ip addr add 192.168.1.2/24 dev wlp2s0
RTNETLINK answers: File exists
pierre@robota:~$ sudo ip addr del 192.168.1.2/24 dev wlp2s0
pierre@robota:~$ ip -4 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
roup default qlen 1
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
UP group default qlen 1000
    inet 192.168.43.244/24 brd 192.168.43.255 scope global dynamic wl
p2s0
        valid_lft 3524sec preferred_lft 3524sec
```

Suppression de l'interface réseau objet de l'opérande cible.

\$ ip link set wlp2s0 up

Activation de l'interface réseau ciblée.

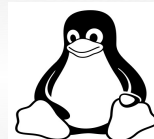
\$ ip link set wlp2s0 down

Désactivation de l'interface réseau ciblée.

\$ ip route add default via 192.168.1.2

```
pierre@robota:~$ sudo ip route add default via 192.168.1.2
pierre@robota:~$ ip -4 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
roup default qlen 1
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
UP group default qlen 1000
    inet 192.168.43.244/24 brd 192.168.43.255 scope global dynamic wl
p2s0
        valid_lft 3166sec preferred_lft 3166sec
    inet 192.168.1.2/24 scope global wlp2s0
        valid_lft forever preferred_lft forever
```

Ajout d'une gateway par défaut.



/etc/network/interfaces :

\$ vim /etc/network/interfaces

Configuration des interfaces réseau.

Après des modifications sur le fichier **interfaces**, relancer les services réseau :

\$ /etc/init.d/networking restart

Pour relancer les périphériques associés au réseau notamment après un chargement de firmware wifi (\$ **modprobe firmware-iwlwifi** *for exemple*) :

\$ /etc/init.d/dbus reload

```
pierre@robota:~$ /etc/init.d/dbus reload  
[ ok ] Reloading dbus configuration (via systemctl): dbus.service.
```



Client wicd :

Sur certaines distributions Linux il arrive que le gestionnaire de connexions réseaux graphiques soit dysfonctionnel.

Il est alors conseillé d'utiliser le client **wicd**, utilisable graphiquement comme en ligne de commande.

\$ apt-get install wicd-cli

Installation.

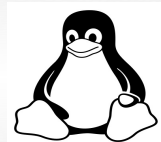
\$ /etc/init.d/wicd start

```
pierre@robota:~$ sudo /etc/init.d/wicd start
[sudo] Mot de passe de pierre :
[ ok ] Starting wicd (via systemctl): wicd.service.
pierre@robota:~$
```

Initialisation.

\$ wicd-cli

Utilisation console.



Quelques outils d'analyse du réseau :

`$ nslookup server_address [(or domain name)]`

Permet d'interroger interactivement les serveurs internet.

```
pierre@robota:~$ nslookup www.google.com
Server:      192.168.43.1
Address:     192.168.43.1#53
```

```
Non-authoritative answer:
```

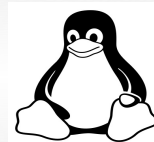
```
Name:   www.google.com
Address: 74.125.206.99
Name:   www.google.com
Address: 74.125.206.105
Name:   www.google.com
Address: 74.125.206.147
Name:   www.google.com
Address: 74.125.206.106
Name:   www.google.com
Address: 74.125.206.103
Name:   www.google.com
Address: 74.125.206.104
```

`$ ping ip [(or domain name)]`

```
pierre@robota:~$ ping -c 4 www.google.com
PING www.google.com (74.125.206.105) 56(84) bytes of data.
64 bytes from wk-in-f105.1e100.net (74.125.206.105): icmp_seq=1 ttl=4
4 time=47.8 ms
64 bytes from wk-in-f105.1e100.net (74.125.206.105): icmp_seq=2 ttl=4
4 time=71.8 ms
64 bytes from wk-in-f105.1e100.net (74.125.206.105): icmp_seq=3 ttl=4
4 time=64.4 ms
64 bytes from wk-in-f105.1e100.net (74.125.206.105): icmp_seq=4 ttl=4
4 time=66.3 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 47.851/62.609/71.825/8.947 ms
```

Requête destinée à déterminer l'activité d'une adresse sur le réseau par envoi de paquets **ICMP**. L'option `-c` sert à définir un compteur en arguments de façon à interrompre la requête **ping**. On peut observer en bas de la capture, le nombre de paquets envoyés, le nombre reçu et le pourcentage de perte ainsi que le délai de réponse entre l'envoi et la réception.

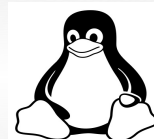


\$ whois ip [(or domain name)]

```
pierre@robota: ~  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
Whois Server Version 2.0  
Domain names in the .com and .net domains can now be registered  
with many different competing registrars. Go to http://www.internic.net  
for detailed information.  
  
Domain Name: GOOGLE.COM  
Registrar: MARKMONITOR INC.  
Sponsoring Registrar IANA ID: 292  
Whois Server: whois.markmonitor.com  
Referral URL: http://www.markmonitor.com  
Name Server: NS1.GOOGLE.COM  
Name Server: NS2.GOOGLE.COM  
Name Server: NS3.GOOGLE.COM  
Name Server: NS4.GOOGLE.COM  
Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited  
Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited  
Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited  
Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited  
Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited  
Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited  
Updated Date: 20-jul-2011  
Creation Date: 15-sep-1997  
Expiration Date: 14-sep-2020  
  
>>> Last update of whois database: Wed, 24 May 2017 16:18:25 GMT <<<  
  
For more information on Whois status codes, please visit https://icann.org/epp  
:  
█
```

Ci-dessus : résultat de sortie de la commande **whois google.com**

whois parvient a obtenir des bases de données des domaines **.com .edu .net .org**.



Netstat :

La commande **netstat** affiche les connexions réseau, les tables de routage, les statistiques des interfaces, les connexions masquées, les messages **netlink**, et les membres **multicast**.

netstat -i

Affiche la table des interfaces noyau.

```
root@robota:~# netstat -i
Table d'interfaces noyau
Iface    MTU      RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s25  1500      0      0      0      0        0      0      0      0  BMU
lo        65536    188      0      0      0       188      0      0      0  LRU
wlp2s0    1500    10718      0      0      0      8999      0      0      0  BMRU
root@robota:~#
```

netstat -ta

Affiche toutes les connexions (-a) des **sockets** TCP (-t).

```
root@robota:~# netstat -ta
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale      Adresse distante      Etat
tcp      0      0 0.0.0.0:xmpp-server  0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:ssh          0.0.0.0:*              LISTEN
tcp      0      0 localhost:ipp        0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:xmpp-client  0.0.0.0:*              LISTEN
tcp      0      0 localhost:mysql      0.0.0.0:*              LISTEN
tcp      0      0 robota:36260         83.114.13.109.rev:https ESTABLISHED
tcp6     0      0 [::]:xmpp-server     [::]:*                 LISTEN
tcp6     0      0 [::]:ssh              [::]:*                 LISTEN
tcp6     0      0 localhost:ipp        [::]:*                 LISTEN
tcp6     0      0 [::]:xmpp-client     [::]:*                 LISTEN
tcp6     0      0 [::]:http             [::]:*                 LISTEN
root@robota:~#
```

On voit les connections en cours (**ESTABLISHED**) et les démons qui attendent une connexion (**LISTEN**).

L'option **-r** affiche la table de routage, telle qu'elle serait affichée par la commande **route** et l'option **-n** force **netstat** à afficher les adresses IP à la place des noms de machines et de réseaux (utile pour éviter la résolution de l'adresse IP en nom lorsque le résolveur de noms n'est pas configuré, et ainsi éviter une attente interminable)

```

root@robota:~# netstat -rn
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic  MSS Fenêtre irtt Iface
0.0.0.0          192.168.231.254 0.0.0.0          UG      0 0      0 wlp2s0
2.2.2.1          192.168.231.254 255.255.255.255 UGH      0 0      0 wlp2s0
169.254.0.0      0.0.0.0          255.255.0.0      U      0 0      0 wlp2s0
192.168.224.0    0.0.0.0          255.255.248.0    U      0 0      0 wlp2s0
root@robota:~#

```

La quatrième colonne affiche un drapeau pour la route : *U* pour une route active (*up*), *H* si la cible est un hôte (*host*), *G* si la route utilise une passerelle (*gateway*), *D* pour une route configurée dynamiquement (*dynamic route*) et *M* pour une route modifiée par le démon de routage ou par **redirect** (*modified*).

La cinquième colonne affiche la taille maximum de segment construit par défaut par le noyau pour les connexions TCP via cette route (*Maximum Segment Size*).

La sixième colonne affiche la taille de fenêtre par défaut pour les connexions TCP via cette route (*Window*).

La septième colonne affiche la valeur *IRTT* (*Initial Round Trip Time*). Le protocole TCP certifie que les données sont délivrées en retransmettant un paquet si il a été perdu. Le protocole TCP compte en permanence de la durée de livraison d'un paquet, pour connaître combien de temps à attendre avant de retransmettre un paquet. C'est ce que l'on appelle le *Round Trip Time*. Le *Initial Round Trip Time* est la valeur utilisée par le protocole TCP lorsqu'une connexion est établie (0 est la valeur par défaut). Pour certains réseaux lents comme les réseaux de radio amateur, le délai est trop court et peut causer des retransmission inutiles.

Source :

<http://www.linux-france.org/~mdecore/linux/doc/memo2/node51.html>

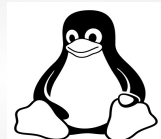
netstat -tupan

Un incontournable de l'administrateur réseau. Voir toutes les connexions actives et les ports des connexions tcp/udp ainsi que les adresses.

```

root@robota:~# netstat -tupan
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale      Adresse distante      Etat      PID/Program name
tcp      0      0 0.0.0.0:5269        0.0.0.0:*              LISTEN    950/luas.1
tcp      0      0 0.0.0.0:22         0.0.0.0:*              LISTEN    620/sshd
tcp      0      0 0.0.0.0:1:631      0.0.0.0:*              LISTEN    1799/cupsd
tcp      0      0 0.0.0.0:5222       0.0.0.0:*              LISTEN    950/luas.1
tcp      0      0 0.0.0.0:1:3306     0.0.0.0:*              LISTEN    641/mysqld
tcp6     0      0 :::5269            :::*                  LISTEN    950/luas.1
tcp6     0      0 :::22              :::*                  LISTEN    620/sshd
tcp6     0      0 :::1:631           :::*                  LISTEN    1799/cupsd
tcp6     0      0 :::5222            :::*                  LISTEN    950/luas.1
tcp6     0      0 :::80              :::*                  LISTEN    953/apache2
udp      0      0 0.0.0.0:631        0.0.0.0:*              1800/cups-browsed
udp      0      0 0.0.0.0:5353       0.0.0.0:*              560/avahi-daemon: r
udp      0      0 0.0.0.0:52511      0.0.0.0:*              560/avahi-daemon: r
udp      0      0 0.0.0.0:68         0.0.0.0:*              694/dhclient
udp6     0      0 :::5353            :::*                  560/avahi-daemon: r
udp6     0      0 :::49136           :::*                  560/avahi-daemon: r
root@robota:~#

```



Wireshark :

Wireshark est un analyseur de paquets libre utilisé dans le dépannage et l'analyse de réseaux informatiques, le développement de protocoles, l'éducation et la rétro-ingénierie.

Wireshark utilise la bibliothèque logicielle **GTK+** pour l'implémentation de son interface utilisateur et **pcap** pour la capture des paquets ; il fonctionne sur de nombreux environnements compatibles UNIX comme GNU/Linux, FreeBSD, NetBSD, OpenBSD ou Mac OSX, mais également sur Microsoft Windows.

Wireshark reconnaît 1 515 protocoles.

Source :

<https://fr.wikipedia.org/wiki/Wireshark>

Wireshark est un outil complexe, puissant et dont les usages sont multiples et variés. Pour de plus amples informations, se référer à la documentation officielle ou au **debian handbook**.

https://www.wireshark.org/docs/wsug_html_chunked/

<https://debian-handbook.info/browse/stable/>



Les outils de routage :

ROUTAGE :

Le **routage** est le mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires. Le routage est une tâche exécutée dans de nombreux réseaux, tels que le réseau téléphonique, les réseaux de données électroniques comme Internet, et les réseaux de transports. Sa performance est importante dans les réseaux décentralisés, c'est-à-dire où l'information n'est pas distribuée par une seule source, mais échangée entre des agents indépendants.

Dans le modèle OSI, le routage s'effectue en examinant les informations situées dans la couche de réseau tels que l'IP.

Source :

<https://fr.wikipedia.org/wiki/Routage>

route -n

Affiche la table de routage ip en adresses numériques.

```
pierre@robota:~$ sudo route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref     U
se Iface
0.0.0.0          192.168.231.254 0.0.0.0          UG        600     0
0 wlp2s0
2.2.2.1          192.168.231.254 255.255.255.255 UGH       600     0
0 wlp2s0
169.254.0.0      0.0.0.0          255.255.0.0      U         1000    0
0 wlp2s0
192.168.224.0    0.0.0.0          255.255.248.0    U         600     0
0 wlp2s0
```

\$ traceroute domain_name [(or ip address)]

```
pierre@robota:~$ traceroute www.google.com
traceroute to www.google.com (74.125.206.106), 30 hops max, 60 byte packets
 1 gateway (192.168.43.1)  5.877 ms  5.865 ms  5.855 ms
 2 172.31.255.250 (172.31.255.250)  24.520 ms  21.751 ms  24.504 ms
 3 172.31.255.10 (172.31.255.10)  31.063 ms  55.391 ms  55.391 ms
 4 pl1-9k-1-bel024.intf.routers.proxad.net (194.149.162.5)  55.386 ms
 55.377 ms  55.369 ms
 5 pl1-9k-1-be2100.intf.routers.proxad.net (194.149.162.29)  55.359
ms  55.352 ms  55.347 ms
 6 pl1-crs16-1-bel004.intf.routers.proxad.net (78.254.249.129)  55.3
28 ms  45.508 ms  42.486 ms
 7 cbv-crs8-1.intf.routers.proxad.net (78.254.249.102)  39.787 ms  3
7.985 ms  37.960 ms
 8 72.14.221.62 (72.14.221.62)  37.952 ms  42.366 ms  42.366 ms
 9 108.170.244.197 (108.170.244.197)  42.364 ms  44.768 ms  108.170.2
45.5 (108.170.245.5)  42.326 ms
10 108.170.235.161 (108.170.235.161)  44.770 ms  209.85.142.191 (209.
85.142.191)  44.739 ms  209.85.142.91 (209.85.142.91)  44.736 ms
11 216.239.51.110 (216.239.51.110)  47.960 ms  216.239.48.37 (216.239
.48.37)  47.933 ms  216.239.48.75 (216.239.48.75)  38.524 ms
12 216.239.47.177 (216.239.47.177)  38.452 ms  66.249.94.159 (66.249.
94.159)  38.452 ms  66.249.94.29 (66.249.94.29)  47.393 ms
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 wk-in-f106.1e100.net (74.125.206.106)  66.577 ms  44.394 ms  35.7
89 ms
```

La commande **traceroute** fournit une sortie décrivant les noms et adresses IP des routeurs successifs, précédés d'un numéro d'ordre et du temps de réponse minimum, moyen et maximum.

Fonctionnement de traceroute :

traceroute appuie son fonctionnement sur le champ TTL des paquets IP. En effet chaque paquet IP possède un champ durée de vie (*TTL, Time To Live*) décrémenté à chaque passage d'un routeur. Lorsque ce champ arrive à zéro, le routeur, considérant que le paquet tourne en boucle, détruit ce paquet et envoie une notification **ICMP** à l'expéditeur.

Ainsi, **traceroute** envoie des paquets à un **port UDP** non privilégié, réputé non utilisé (le port 33434 par défaut) avec un TTL valant 1. Le premier routeur rencontré va supprimer le paquet et renvoyer un paquet **ICMP** donnant notamment l'adresse IP du routeur ainsi que le temps de propagation en boucle. **traceroute** va ainsi incrémenter séquentiellement le champ durée de vie, de manière à obtenir une réponse de chacun des routeurs sur le chemin, jusqu'à obtenir une réponse «*ICMP port unreachable*» de la part de la machine cible.

Source :

<http://www.commentcamarche.net/contents/715-traceroute>



Localhost :

Localhost est le nom donné pour se référer à l'ordinateur local. Il possède une adresse IPV4 : « 127.0.0.1 » et une adresse IPV6 : « ::1 » aussi appelé adresse **loopback**. Cette adresse est utilisée par la machine pour s'interroger elle-même. Avec un **ping** par exemple :

```
root@robota:~# ping localhost
PING localhost(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.035 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.067 ms
64 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from localhost (::1): icmp_seq=5 ttl=64 time=0.066 ms
64 bytes from localhost (::1): icmp_seq=6 ttl=64 time=0.064 ms
^C
--- localhost ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5126ms
rtt min/avg/max/mdev = 0.035/0.062/0.073/0.013 ms
root@robota:~# ping -c 4 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.038 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.038/0.040/0.045/0.008 ms
root@robota:~# █
```

```
root@robota:~# ping6 localhost
PING localhost(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.064 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.064 ms
64 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.063 ms
^C
--- localhost ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3073ms
rtt min/avg/max/mdev = 0.040/0.057/0.064/0.013 ms
root@robota:~# █
```

De la même manière l'adresse **localhost** servira pour accéder aux fichiers de la machine hôte sur un serveur web local apache par exemple :





Pour effectuer ce test, placer des fichiers lisibles par un navigateur (html, php, etc.) dans `/var/www/`.



BIND 9 :

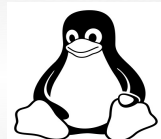
La mise en place d'un serveur DNS sur un réseau permet de remplacer les adresses IP des machines par un nom. Ainsi, il est même possible d'associer plusieurs noms à la même machine pour mettre en évidence les différents services possibles. Du coup, `www.example.com` et `pop.example.com`, peuvent pointer sur le serveur principal où sont présents le serveur de mail et l'intranet de l'entreprise dont le domaine serait `example.com`. C'est tout de même plus facile que de se rappeler que ces deux services tournent sur la machine dont l'adresse IP est `192.168.0.1`.

Source :

<https://wiki.debian.org/fr/Bind9>

Nous ne traiterons pas de Bind9 dans cet ouvrage destiné aux néophytes et utilisateurs intermédiaires car la mise en place et la configuration de cet outil nécessitent des connaissances approfondies en système et réseaux. Cependant, dans ce chapitre réservé à l'administration du réseau sous Debian il nous a paru important de mentionner cet outil essentiel.

Une documentation importante existe sur le sujet.



Désactivation de l'ipv6 :

Pour certaines raisons, on peut vouloir désactiver l'ipv6. Soit momentanément, à des fins de tests ou pour une action ciblée soit de façon permanente.

Désactivation momentanée :

On vérifie le statut de l'ipv6 :

```
$ cat /proc/sys/net/ipv6/conf/all/disable_ipv6
```

Le résultat à la sortie est un booléen binaire : 1 ou 0.

0 si l'IPv6 n'est pas désactivée, 1 dans le cas contraire.

```
pierre@robota:~$ cat /proc/sys/net/ipv6/conf/all/disable_ipv6
0
```

Ici l'ipv6 est donc bien activée.

Pour la désactiver jusqu'au prochain **reboot** de la machine

```
$ echo «1» > /proc/sys/net/ipv6/conf/all/disable_ipv6
```

```
root@robota:~# echo "1" > /proc/sys/net/ipv6/conf/all/disable_ipv6
```

On écrase le fichier **disable_ipv6** pour le réécrire avec le contenu «1».

L'IPv6 est maintenant désactivée. On peut le vérifier à l'aide de la commande **cat**.

Désactivation permanente :

Pour désactiver l'ipv6 d facon permanente on peut modifier le fichier **/etc/sysctl.conf**

On y ajoute les lignes suivantes :

```
#####
# Disable IPV6
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
```

Puis on exécute

\$ sysctl -p

```
pierre@robota:/etc$ sudo sysctl -p
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
pierre@robota:/etc$
```

Pour appliquer les changements.

```
pierre@robota:/etc$ ip -6 addr show
pierre@robota:/etc$ ip -4 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
roup default qlen 1
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
UP group default qlen 1000
    inet 192.168.1.25/24 brd 192.168.1.255 scope global dynamic wlp2s
0
        valid_lft 42827sec preferred_lft 42827sec
```

On peut ici observer que la commande **ip -6 addr show** ne retourne rien dans la console graphique. Pour cause, l'**ipv6** n'est plus actif.

La désactivation est alors permanente sauf modification du fichier.

VI. Administration des comptes, logs et processus

\$ w

```
pierre@robota:~/Test$ w
23:30:53 up 4:49, 1 user, load average: 0,03, 0,06, 0,07
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU   WHAT
pierre    tty2     :0            18:43       4:48m  3:53   0.00s  single
pierre@robota:~/Test$
```

Signifie « who is logged and what he's doing ». Donne des informations sur la session ouverte et le processus courant.

\$ groups

Affiche les groupes auquel appartient l'utilisateur courant.

On peut aussi préciser l'utilisateur en paramètres :

\$ groups user

```
pierre@robota:~/Test$ groups pierre
pierre : pierre cdrom floppy sudo audio dip video plugdev netdev bluetooth lpadmin scanner
pierre@robota:~/Test$ groups
pierre cdrom floppy sudo audio dip video plugdev netdev bluetooth lpadmin scanner
pierre@robota:~/Test$
```

\$ who

Means « who is logged and where ».

```
pierre@robota:~/Test$ who
pierre    tty2                2017-05-19 18:43 (:0)
```

\$ whoami

Means « who is logged ».

```
pierre@robota:~/Test$ who
pierre    tty2                2017-05-19 18:43 (:0)
```

\$ passwd user

Changer le password de l'utilisateur ciblé.

```
pierre@robota:~/Test$ passwd pierre
Changement du mot de passe pour pierre.
Mot de passe UNIX (actuel) :
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
```



\$ sudo -s / sudo -i

```
pierre@robota:~/Test$ sudo -i
[sudo] Mot de passe de pierre :
root@robota:~# sudo -s
root@robota:~#
```

Permet de devenir « superutilisateur », soit **root**.

\$ su user

```
root@robota:~# su testuser
testuser@robota:/root$
```

Permet de devenir l'utilisateur ciblé.

\$ exit

```
root@robota:~# exit
déconnexion
pierre@robota:~/Test$
```

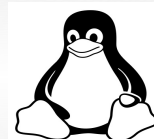
Clôt la connexion.

Note : Cependant, à chaque fois que l'on initie une connexion utilisateur temporaire depuis sa propre session, la session reste inchangée. La préemption et usurpation du compte cible ne valent qu'en console à un moment t.

\$ adduser new_user

```
root@robota:/home/pierre# adduser usertest
Ajout de l'utilisateur « usertest » ...
Ajout du nouveau groupe « usertest » (1003) ...
Ajout du nouvel utilisateur « usertest » (1002) avec le groupe « user
test » ...
Le répertoire personnel « /home/usertest » existe déjà. Rien n'est c
opié depuis « /etc/skel ».
adduser : Attention ! Le répertoire personnel « /home/usertest » n'ap
partient pas à l'utilisateur que vous êtes en train de créer.
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
Changing the user information for usertest
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Cette information est-elle correcte ? [0/n]0
```

Création d'un compte utilisateur.



Dans l'exemple ci-dessus on observe que le répertoire personnel du nouvel utilisateur est déjà existant. En cause, on avait précédemment créé un autre utilisateur du même nom, puis supprimé sans ajouter d'option pour la suppression du répertoire personnel. Par défaut et sans options passées en paramètres, **adduser nom_user** crée le répertoire personnel (/home/user) de l'utilisateur au même nom.

Nous sommes ici dans une situation de conflit à des fins de test. Pour pallier à ce problème on va donc supprimer utilisateur ainsi que son répertoire personnel et en recréer un nouveau pour lequel on spécifiera un /home/ d'un nom différent :

\$ deluser usertest --group --remove-home --remove-all-files

On va ici supprimer l'utilisateur « usertest » ainsi que son /home et boîte mail, tous les fichiers du système qui lui appartenaient ainsi le groupe d'appartenance du même nom. La commande envoi un code de retour compris entre 1 et 9. Se référer à man.

\$ sudo commande

L'usage de nombreuses commandes, comme les tâches d'administration ou même la visualisation du contenu des répertoires système, nécessitent les droits et privilèges du superutilisateur, soit **root**.

Parfois, on ne souhaite pas forcément s'identifier en tant que **root** ou en tant qu'un autre utilisateur aux privilèges plus élevés, on va alors passer par la commande **sudo**, qui va nous permettre de disposer momentanément des droits du superutilisateur tout en restant connecté sur notre session courante.

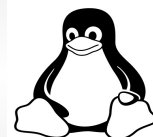
```
pierre@robota:~$ apt-get install latex
E: Impossible d'ouvrir le fichier verrou /var/lib/dpkg/lock - open (13: Permission non accordée)
E: Impossible de verrouiller le répertoire d'administration (/var/lib/dpkg/). Avez-vous les privilèges du superutilisateur ?
pierre@robota:~$
```

Dans l'exemple ci-dessus, l'installation du paquet nécessite des droits. Ne les possédant pas par défaut, la commande nous retourne une erreur et l'impossibilité de s'exécuter. Nous allons donc utiliser **sudo**.

Pour utiliser cette commande il est parfois nécessaire de l'installer en **root** :

apt-get install sudo

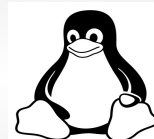
Le plus souvent, cette commande est native. Une fois installée il faut ajouter l'utilisateur à qui l'on veut octroyer les droits d'exécution de **sudo** en l'ajoutant dans le groupe **sudo**. Ce qui nous permet par ailleurs de découvrir la création/suppression de groupe ainsi que l'ajout d'un



utilisateur dans ces groupes.

```
# addgroup user_test sudo
```

```
pierre@robota: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
root@robota:~# adduser user_test
Ajout de l'utilisateur « user_test » ...
Ajout du nouveau groupe « user_test » (1003) ...
Ajout du nouvel utilisateur « user_test » (1002) avec le groupe « user_test » ...
Création du répertoire personnel « /home/user_test »...
Copie des fichiers depuis « /etc/skel »...
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
Changing the user information for user_test
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Cette information est-elle correcte ? [0/n]0
root@robota:~# addgroup test
Ajout du groupe « test » (GID 1004)...
Fait.
root@robota:~# addgroup user_test sudo test
addgroup : Un ou deux noms maximum.
root@robota:~# addgroup user_test sudo
Ajout de l'utilisateur « user_test » au groupe « sudo »...
Adding user user_test to group sudo
Fait.
root@robota:~# addgroup user_test test
Ajout de l'utilisateur « user_test » au groupe « test »...
Adding user user_test to group test
Fait.
root@robota:~#
```



Dans cet exemple on a créé l'utilisateur **user_test**, on a créé un groupe **test** et ajouté **user_test** au groupe **sudo** et **test**.

Pour que les modifications soient prises en compte et la commande **sudo** utilisable par l'utilisateur il faut redémarrer la session.

On peut ensuite l'utiliser.

```
$ sudo apt-get install paquet
```

```
pierre@robota: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
pierre@robota:~$ sudo apt-get install latex-mk
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
The following additional packages will be installed:
 fonts-lato fonts-lmodern libfile-homedir-perl libfile-which-perl
 libpotrace0 libptexenc1 libruby2.3 libsyntax1 libtexlua52
 libtexluajit2 libyaml-tiny-perl libzip-0-13 lmodern prosper
 ps2eps rake ruby ruby-did-you-mean ruby-minitest ruby-net-telnet
 ruby-power-assert ruby-test-unit ruby2.3 rubygems-integration
 tlutils tex-common texlive-base texlive-binaries
 texlive-extra-utils texlive-font-utils
 texlive-generic-recommended texlive-latex-base
 texlive-latex-base-doc texlive-latex-recommended
 texlive-latex-recommended-doc texlive-pictures
 texlive-pictures-doc texlive-pstricks texlive-pstricks-doc
Paquets suggérés :
 graphicsmagick imagemagick-compat gv hevea latex2rtf transfig ri
 ruby-dev bundler perl-tk chktex dvidvi dvipng fragmaster lacheck
 latexdiff latexmk purifyeps xindy psutils dot2tex prerex
 ruby-tcltk | libtcltk-ruby texlive-latex-extra
Les NOUVEAUX paquets suivants seront installés :
 fonts-lato fonts-lmodern latex-mk libfile-homedir-perl
 libfile-which-perl libpotrace0 libptexenc1 libruby2.3 libsyntax1
 libtexlua52 libtexluajit2 libyaml-tiny-perl libzip-0-13 lmodern
 prosper ps2eps rake ruby ruby-did-you-mean ruby-minitest
 ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.3
 rubygems-integration tlutils tex-common texlive-base
 texlive-binaries texlive-extra-utils texlive-font-utils
 texlive-generic-recommended texlive-latex-base
 texlive-latex-base-doc texlive-latex-recommended
 texlive-latex-recommended-doc texlive-pictures
 texlive-pictures-doc texlive-pstricks texlive-pstricks-doc
0 mis à jour, 40 nouvellement installés, 0 à enlever et 67 non mis à
jour.
Il est nécessaire de prendre 543 Mo/543 Mo dans les archives.
Après cette opération, 793 Mo d'espace disque supplémentaires seront
utilisés.
Souhaitez-vous continuer ? [0/n]
```

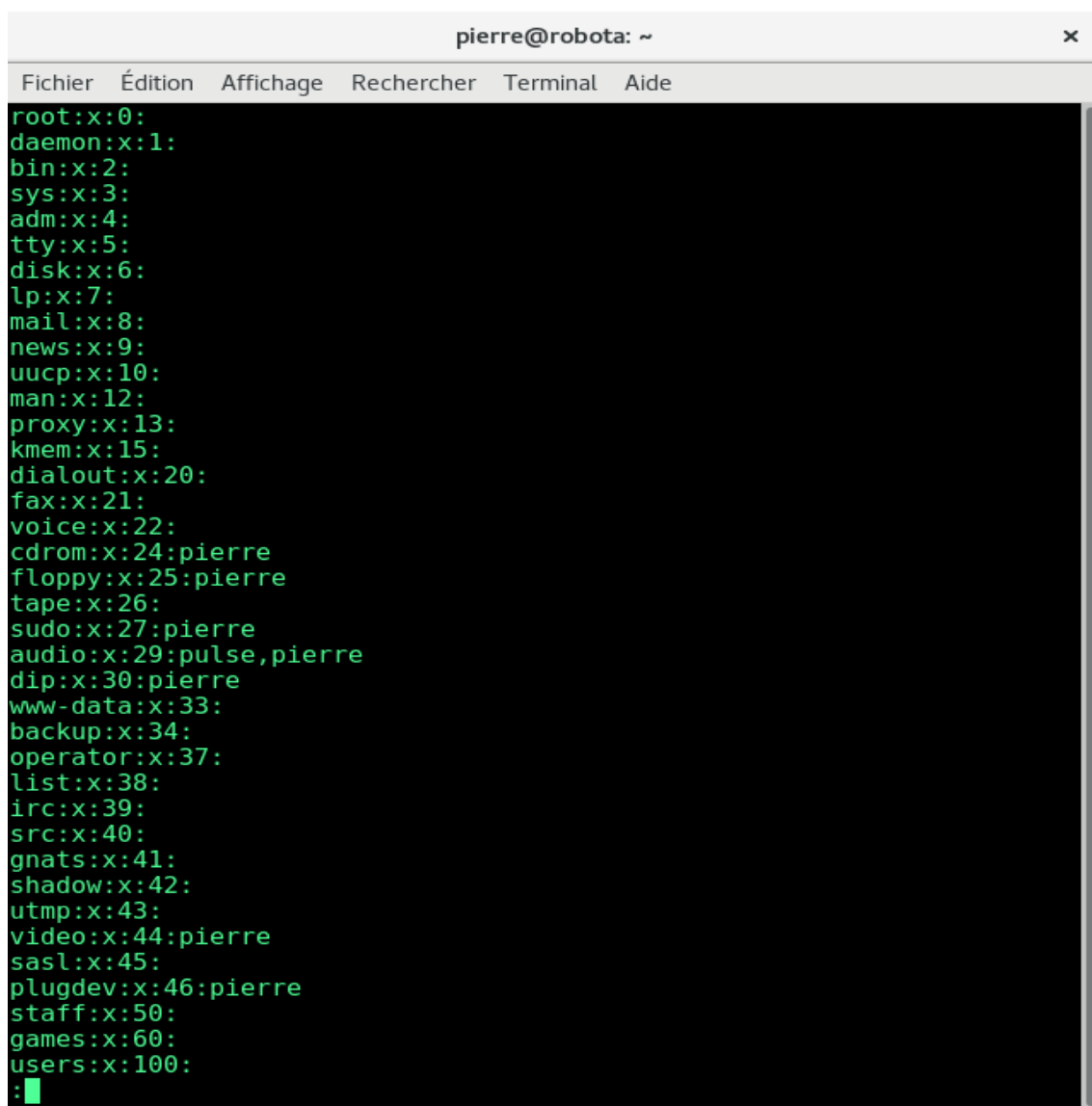
Dans l'exemple ci-dessus, on a utilisé **sudo** avec **pierre** et non **user_test** (déjà supprimé) pour des raisons pratiques.

```
pierre@robota:~$ sudo delgroup user_test test
[sudo] Mot de passe de pierre :
Suppression de l'utilisateur « user_test » du groupe « test »...
Fait.
pierre@robota:~$
```

\$ sudo delgroup user group

On peut visualiser les groupes présents sur le système par la commande

\$ cat /etc/group | less



```
pierre@robota: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:pierre
floppy:x:25:pierre
tape:x:26:
sudo:x:27:pierre
audio:x:29:pulse,pierre
dip:x:30:pierre
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:pierre
sasl:x:45:
plugdev:x:46:pierre
staff:x:50:
games:x:60:
users:x:100:
:
```

\$ date

Permet de régler ou d'afficher la date. Pour les options, se référer à **man**.

\$ ps

Affiche la liste des processus actifs dans la console.

L'option **-a** est à préciser pour visualiser tous les processus actifs dans la session courante ouverte sur la machine.

```
pierre@robota: ~  
Fichier  Édition  Affichage  Rechercher  Terminal  Aide  
pierre@robota:~$ ps  
  PID TTY          TIME CMD  
 1915 pts/0    00:00:00 bash  
 4884 pts/0    00:00:00 ps  
pierre@robota:~$ ps -a  
  PID TTY          TIME CMD  
 1196 tty1      00:00:00 gnome-session-b  
 1225 tty1      00:00:06 gnome-shell  
 1257 tty1      00:00:00 Xwayland  
 1291 tty1      00:00:00 gnome-settings-  
 1379 tty2      00:01:04 Xorg  
 1388 tty2      00:00:00 gnome-session-b  
 1475 tty2      00:01:38 gnome-shell  
 1568 tty2      00:00:02 gnome-settings-  
 1601 tty2      00:00:00 tracker-miner-a  
 1608 tty2      00:00:00 gsd-printer  
 1610 tty2      00:00:00 tracker-miner-f  
 1617 tty2      00:00:02 owncloud  
 1624 tty2      00:00:00 tracker-miner-u  
 1625 tty2      00:00:01 gnome-software  
 1626 tty2      00:00:01 tracker-extract  
 1730 tty2      00:00:00 sh  
 1731 tty2      00:00:00 pxgsettings  
 1749 tty2      00:00:00 single <defunct>  
 4885 pts/0    00:00:00 ps  
pierre@robota:~$ █
```

\$ kill -s SIGNAL PID

```
root@robota:~# kill -s SIGTERM 1617
root@robota:~# ps -a
  PID TTY          TIME CMD
 1196 tty1        00:00:00 gnome-session-b
 1225 tty1        00:00:06 gnome-shell
 1257 tty1        00:00:00 Xwayland
 1291 tty1        00:00:00 gnome-settings-
 1379 tty2        00:01:06 Xorg
 1388 tty2        00:00:00 gnome-session-b
 1475 tty2        00:01:42 gnome-shell
 1568 tty2        00:00:02 gnome-settings-
 1601 tty2        00:00:00 tracker-miner-a
 1608 tty2        00:00:00 gsd-printer
 1610 tty2        00:00:00 tracker-miner-f
 1624 tty2        00:00:00 tracker-miner-u
 1625 tty2        00:00:01 gnome-software
 1626 tty2        00:00:01 tracker-extract
 1749 tty2        00:00:00 single <defunct>
 4908 pts/0        00:00:00 sudo
 4909 pts/0        00:00:00 bash
 4924 pts/0        00:00:00 ps
root@robota:~#
```

Permet d'envoyer un signal à un processus. Par défaut c'est un signal **SIGTERM**, signifiant : « tuer, mettre fin » au processus ciblé. Où PID est le numéro de l'identifiant du processus. Il est nécessaire de disposer des droits **root** ou de passer par **sudo** pour utiliser la commande **kill**.

Astuce : \$ sudo !! exécute la dernière commande en root.

Exemple :

```
$ rmdir /rep_system // forbidden
```

```
$ sudo !! // sudo rmdir /rep_system
```


\$ top

Affiche les processus courants. Des options peuvent être renseignés pour filtrer les informations ou les indiquer en temps réel ou donné.

Par exemple : `$ top -u pierre [-d](optionnel) 1`

```
top - 23:59:07 up 5:17, 1 user, load average: 0,02, 0,14, 0,12
Tasks: 206 total, 1 running, 204 sleeping, 0 stopped, 1 zombie
%Cpu(s): 1,0 us, 0,2 sy, 0,0 ni, 97,8 id, 1,0 wa, 0,0 hi, 0,0 s
KiB Mem : 3844004 total, 1765024 free, 1024828 used, 1054152 buff
KiB Swap: 5119996 total, 5119996 free, 0 used. 2377048 avai
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
1426	pierre	20	0	2422252	262948	62516	S	2,9	6,8	3:25.02
6098	pierre	20	0	44920	3828	3176	R	2,0	0,1	0:00.28
1715	pierre	20	0	601028	33460	24988	S	1,0	0,9	0:06.06
1319	pierre	20	0	65116	6732	5584	S	0,0	0,2	0:00.08
1320	pierre	20	0	234876	2328	24	S	0,0	0,1	0:00.00
1326	pierre	20	0	213320	6456	5720	S	0,0	0,2	0:00.08
1329	pierre	20	0	201292	5688	5168	S	0,0	0,1	0:00.00
1331	pierre	20	0	382940	72076	37324	S	0,0	1,9	2:03.89
1337	pierre	20	0	45840	4608	3396	S	0,0	0,1	0:00.62
1339	pierre	20	0	547152	12568	10616	S	0,0	0,3	0:00.12
1397	pierre	20	0	11084	332	0	S	0,0	0,0	0:00.02

affichera les processus appartenant à pierre toutes les secondes.

top est une commande très riche. Se référer à **man** pour visualiser ses nombreuses options.

\$ htop

Contrairement à **top**, **htop** n'est pas natif (n.d.a. *natif* signifie installé avec le système de base) et nécessite d'être installé via **apt** ou **aptitude**. Il permet un monitoring des processus en indiquant graphiquement dans le terminal la charge software/hardware.

Il dispose lui aussi de nombreuses options.

```
1  [ | 0.7%] Tasks: 110, 264 thr; 1 running
2  [ | 0.7%] Load average: 0.07 0.10 0.09
3  [ | 0.0%] Uptime: 05:23:32
4  [ | 4.0%]
Mem[|||||||1.19G/3.67G]
Swp[|0K/4.88G]

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
1426 pierre    20   0 2368M  261M  62520 S   2.7  7.0   3:36.59 /usr/b
1331 pierre    20   0  382M  73676 38920 S   1.3  1.9   2:12.49 /usr/l
6250 pierre    20   0 24648  3660   2928 R   0.7  0.1   0:00.13 htop
1528 pierre    20   0 1376M  40084 31280 S   0.7  1.0   0:01.43 /usr/l
1817 pierre    20   0 4808M  283M  144M S   0.0  7.6   3:20.78 /usr/l
1518 pierre    20   0 1376M  40084 31280 S   0.0  1.0   0:04.04 /usr/l
1599 pierre    20   0 1091M  73748 57108 S   0.0  1.9   0:00.37 /usr/b
  708 messagebu 20   0  46696  5332   3400 S   0.0  0.1   0:09.52 /usr/b
1336 pierre    20   0  382M  73676 38920 S   0.0  1.9   0:07.38 /usr/l
2103 pierre    20   0 4808M  283M  144M S   0.0  7.6   0:00.13 /usr/l
2105 pierre    20   0 4808M  283M  144M S   0.0  7.6   0:00.12 /usr/l
1715 pierre    20   0  586M  33676 25108 S   0.0  0.9   0:06.52 /usr/l
1176 root        20   0  181M  14600  5240 S   0.0  0.4   0:05.84 /usr/b
   1 root        20   0  200M   7336  5164 S   0.0  0.2   0:02.80 /sbin/
  382 root        20   0 55944   7016  6304 S   0.0  0.2   0:00.94 /lib/s
  416 root        20   0 99004   1552  1336 S   0.0  0.0   0:00.00 /sbin/
  418 root        20   0 47548   5144  2816 S   0.0  0.1   0:00.74 /lib/s
```


ATOP :

Atop est un outil de monitoring interactif du système et des processus.

Il dispose de nombreuses options et les résultats peuvent être exploités graphiquement avec le logiciel **graffana**. Nous ne traiterons pas de **graffana** ici mais une documentation riche et fournie existe sur le sujet.

\$ atop 10 99

La commande **atop** va s'exécuter pendant 90 secondes et s'actualiser toutes les 10 secondes.

ATOP - robota 2017/05/26 12:57:53 ----- 10s elapsed									
PRC	sys	0.20s	user	0.92s	#proc	215	#exit	2	
CPU	sys	3%	user	10%	idle	386%	wait	2%	
cpu	sys	2%	user	7%	idle	92%	cpu001 w	0%	
cpu	sys	1%	user	2%	idle	98%	cpu000 w	0%	
cpu	sys	0%	user	1%	idle	98%	cpu002 w	1%	
cpu	sys	0%	user	0%	idle	98%	cpu003 w	1%	
CPL	avg1	0.06	avg5	0.22	csw	11733	intr	8467	
MEM	tot	3.7G	free	650.5M	buff	216.5M	slab	644.9M	
SWP	tot	4.9G	free	4.9G	vmcom	4.8G	vmlim	6.7G	
DSK	sda		busy	1%	read	0	write	12	
NET	transport		tcpo	10	udpi	0	udpo	0	
NET	network		ipo	10	ipfrw	0	deliv	10	
NET	wlp2s0	0%	pcki	11	pcko	10	so	1 Kbps	

PID	SYSCPU	USRCPU	VGROW	RGROW	RDDSK	WRDSK	CPU	CMD	1/2
2645	0.04s	0.74s	352K	60K	0K	0K	8%	gnome-shell	
2474	0.07s	0.06s	-16K	0K	0K	0K	1%	Xorg	
5086	0.03s	0.03s	676K	604K	0K	0K	1%	atop	
3108	0.01s	0.01s	0K	0K	0K	0K	0%	firefox-esr	
964	0.01s	0.01s	0K	0K	0K	0K	0%	wicd	
3313	0.00s	0.01s	0K	0K	0K	8K	0%	soffice.bin	
640	0.01s	0.00s	0K	0K	0K	0K	0%	mysqld	
996	0.00s	0.01s	0K	0K	0K	0K	0%	gnome-shell	
2767	0.00s	0.01s	0K	0K	0K	16K	0%	owncloud	
2735	0.00s	0.01s	0K	4K	0K	0K	0%	gnome-settings	
959	0.00s	0.01s	0K	0K	0K	0K	0%	lua5.1	
554	0.00s	0.01s	0K	0K	0K	0K	0%	dbus-daemon	
7	0.00s	0.01s	0K	0K	0K	0K	0%	rcu_sched	
321	0.01s	0.00s	0K	0K	0K	0K	0%	irq/27-iwlwifi	
3715	0.01s	0.00s	0K	0K	0K	0K	0%	kworker/2:3	
3805	0.01s	0.00s	0K	0K	0K	0K	0%	kworker/0:1	
2889	0.00s	0.00s	0K	0K	0K	0K	0%	gajim	
2780	0.00s	0.00s	0K	0K	0K	0K	0%	gnome-software	
3604	0.00s	0.00s	0K	0K	0K	0K	0%	eog	
2868	0.00s	0.00s	0K	0K	0K	0K	0%	gnome-terminal	
572	0.00s	0.00s	0K	0K	0K	0K	0%	NetworkManager	
999	0.00s	0.00s	0K	0K	0K	0K	0%	wicd-monitor	
1060	0.00s	0.00s	0K	0K	0K	0K	0%	packagekitd	

Pour utiliser **atop** interactivement sans passer les options en paramètres lorsque l'on tape la commande, on peut taper chaque lettre correspondant à l'option tandis que le processus tourne dans la console.

g = affichage de la vue générique

m = affichage des informations concernant la mémoire

PID	VSIZE	RSIZE	PSIZE	VGROW	RGROW	SWAPSZ	MEM	CMD	1/2
3313	4.6G	313.8M	0K	0K	0K	0K	8%	soffice.bin	
3108	1.9G	282.9M	0K	0K	0K	0K	8%	firefox-esr	
2645	2.4G	245.6M	0K	288K	4K	0K	7%	gnome-shell	
640	1.2G	135.1M	0K	0K	0K	0K	4%	mysqld	

d = affichage des informations du disque

PID	TID	RDDSK	WRDSK	WCANCL	DSK	CMD	1/2
198	-	0K	136K	0K	72%	jbd2/sda4-8	
2775	-	0K	36K	0K	19%	tracker-store	
2767	-	0K	16K	0K	9%	owncloud	
2645	-	0K	0K	0K	0%	gnome-shell	

n = affichage des informations du réseau

c = affiche les commandes à l'origine du lancement des processus

PID	TID	S	CPU	COMMAND-LINE (horizontal scroll with <- and -1/2
2645	-	S	8%	/usr/bin/gnome-shell
2474	-	S	2%	/usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/u
3091	-	S	1%	/usr/bin/nautilus --gapplication-service
5418	-	R	1%	atop 10 99
3313	-	S	0%	/usr/lib/libreoffice/program/soffice.bin --write
3108	-	S	0%	/usr/lib/firefox-esr/firefox-esr

o = affiche les utilisateurs concernés par les processus

u = affiche l'activité des processus par utilisateur

NPROCS	SYSCPU	USRCPU	RSIZE	PSIZE	RDDSK	WRDSK	SNET	CPU	RUID	1/1
13	0.13s	0.80s	1.2G	0K	0K	0K	0	9%	pierre	
15	0.08s	0.05s	89752K	0K	0K	0K	0	1%	root	
1	0.00s	0.01s	5228K	0K	0K	0K	0	0%	messageb	
1	0.00s	0.00s	135.1M	0K	0K	0K	0	0%	mysql	
1	0.00s	0.00s	118.0M	0K	0K	0K	0	0%	Debian-g	
1	0.00s	0.00s	8516K	0K	0K	0K	0	0%	prosody	
1	0.00s	0.00s	2864K	0K	0K	0K	0	0%	rtkit	

C = affiche les informations CPU

Certains modules doivent être activés dans le **kernel** comme **netatop**.

Se référer à **man** pour une liste plus exhaustive des options.



La commande watch :

Cette commande permet l'exécution en boucle ou définie d'une commande passée en paramètre.

\$ watch free

Exécute en boucle la commande **free** avec un intervalle de 2 secondes (temps d'intervalle par défaut de la commande **watch**).

\$ watch -n 10 free -m

Exécute en boucle la commande **free -m** (indique la mémoire utilisée en megabytes toutes les 10 secondes. On peut préciser l'intervalle **n** secondes.

```
Every 10,0s: free -m                      robota: Wed May 31 14:00:28 2017
               total      used      free      shared  buff/cache
Mem:          3753        997       1953         141         802
      2393
Swap:         4999          0       4999
```

\$ watch -n 1 'netstat -an | grep «:443 »'

Exécute toutes les secondes la commande **netstat -an** en listant toutes les connexions établies ou écoutées sur le **port 443** (pour TCP : **HTTPS**, soit **http** avec **SSL - Secure Socket Layer -**).

\$ watch -n 1 'netstat -an | grep «:80»' | wc -l

Liste chaque seconde l'état ou le nombre de connexions sur le **port 80** (**HTTP**).

```
Every 1,0s: netstat -an | grep "...  robota: Wed May 31 13:51:33 2017
1
```

Pour interrompre l'exécution de la commande : **CTRL + C**.

Les principaux fichiers de logs :

/var/log/auth/log

Les logs d'authentification.

```
May 31 22:00:27 robota sudo: pierre : TTY=pts/0 ; PWD=/home/pierre  
; USER=root ; COMMAND=/bin/cat /var/log/kern.log  
May 31 22:00:27 robota sudo: pam_unix(sudo:session): session opened f  
or user root by (uid=0)  
May 31 22:00:27 robota sudo: pam_unix(sudo:session): session closed f  
or user root  
May 31 22:01:43 robota sudo: pierre : TTY=pts/0 ; PWD=/home/pierre  
; USER=root ; COMMAND=/bin/cat /var/log/auth.log  
May 31 22:01:43 robota sudo: pam_unix(sudo:session): session opened f  
or user root by (uid=0)  
pierre@robota:~$
```

/var/log/Xorg.0.log

Les logs du serveur X (serveur graphique)

```
296 1344 1408 800 801 804 816 -hsync -vsync (49.0 kHz eP)  
[ 65.171] (II) modeset(0): Modeline "1280x800"x0.0 60.96 1280 1  
328 1360 1478 800 803 809 825 -hsync -vsync (41.2 kHz e)  
[ 68.748] (II) modeset(0): EDID vendor "LEN", prod id 16401  
[ 68.748] (II) modeset(0): Printing DDC gathered Modelines:  
[ 68.748] (II) modeset(0): Modeline "1280x800"x0.0 68.94 1280 1  
296 1344 1408 800 801 804 816 -hsync -vsync (49.0 kHz eP)  
[ 68.748] (II) modeset(0): Modeline "1280x800"x0.0 60.96 1280 1  
328 1360 1478 800 803 809 825 -hsync -vsync (41.2 kHz e)  
[ 77.066] (II) Axis 0x1 value 1184 is outside expected range [1237  
, 4980]  
See https://wayland.freedesktop.org/libinput/doc/1.6.3//absolute\_coordinate\_ranges.html for details  
[ 844.806] (II) modeset(0): EDID vendor "LEN", prod id 16401  
[ 844.806] (II) modeset(0): Printing DDC gathered Modelines:  
[ 844.806] (II) modeset(0): Modeline "1280x800"x0.0 68.94 1280 1  
296 1344 1408 800 801 804 816 -hsync -vsync (49.0 kHz eP)  
[ 844.806] (II) modeset(0): Modeline "1280x800"x0.0 60.96 1280 1  
328 1360 1478 800 803 809 825 -hsync -vsync (41.2 kHz e)  
[ 1008.941] (II) UnloadModule: "libinput"  
[ 1008.941] (II) UnloadModule: "libinput"  
[ 1008.941] (II) UnloadModule: "libinput"  
[ 1008.941] (II) UnloadModule: "libinput"  
[ 1008.941] (II) UnloadModule: "libinput"  
[ 1008.941] (II) UnloadModule: "libinput"  
[ 1008.941] (II) UnloadModule: "libinput"  
[ 1008.941] (II) UnloadModule: "libinput"  
[ 1009.600] (II) Server terminated successfully (0). Closing log fil  
e.
```


~/xsession-errors

Les logs relatifs à la dernière session graphique et ses erreurs

```
dbus-update-activation-environment: setting SHELL=/bin/bash
dbus-update-activation-environment: setting QT_ACCESSIBILITY=1
dbus-update-activation-environment: setting GDMSESSION=gnome
dbus-update-activation-environment: setting GJS_DEBUG_OUTPUT=stderr
dbus-update-activation-environment: setting GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
dbus-update-activation-environment: setting PWD=/home/pierre/Téléchargements/VNC-6.0.2-Linux-x64
dbus-update-activation-environment: setting XDG_DATA_DIRS=/usr/share/mate:/usr/share/gnome:/usr/local/share/:/usr/share/
dbus-update-activation-environment: setting VTE_VERSION=4601
/etc/X11/Xsession: 26: /etc/X11/Xsession.d/98vboxadd-xclient: notify-send: not found
x-session-manager[3429]: WARNING: Failed to acquire org.gnome.SessionManager
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.
```

/var/log/kern.log

Les logs du Kernel.

/var/log/cron.log

Les logs de Cron

/var/log/messages | /var/log/syslog

```
May 31 21:42:40 robota tracker-extract[1478]: unable to create file '/run/user/1000/dconf/user': Permission non accordée. dconf will not work properly.
May 31 21:42:56 robota kernel: [ 5143.149129] perf: interrupt took too long (4962 > 4950), lowering kernel.perf_event_max_sample_rate to 40250
May 31 21:43:25 robota wpa_supplicant[677]: wlp2s0: WPA: Group rekeying completed with 68:a3:78:df:3c:f0 [GTK=CCMP]
May 31 21:53:25 robota wpa_supplicant[677]: wlp2s0: WPA: Group rekeying completed with 68:a3:78:df:3c:f0 [GTK=CCMP]
May 31 22:01:40 robota kernel: [ 6266.952292] [drm:ironlake_irq_handler [i915]] *ERROR* CPU pipe A FIFO underrun
May 31 22:01:40 robota kernel: [ 6266.952364] [drm:ironlake_irq_handler [i915]] *ERROR* PCH transcoder A FIFO underrun
May 31 22:01:41 robota tracker-extract[1478]: unable to create file '/run/user/1000/dconf/user': Permission non accordée. dconf will not work properly.
```



Les messages généralistes et les logs système.

Ci-dessus, extrait de `/var/log/syslog`.

dmesg

Cette commande fournit une vue non exhaustive des messages du **kernel** - c'est donc un raccourci vers `/var/log/kern.log` - (donc du système du point de vue de la liaison entre le matériel et les logiciels qui l'utilisent). On peut par exemple y voir les informations relatives à la connexion de la carte réseau (dans la capture qui suit une carte wifi).

```
[ 1471.804462] wlp2s0: associated
[ 1471.804538] IPv6: ADDRCONF(NETDEV_CHANGE): wlp2s0: link becomes ready
[ 1471.935797] wlp2s0: Limiting TX power to 20 (20 - 0) dBm as advertised by 20:a6:80:a9:d2:74
[ 1484.486936] wlp2s0: deauthenticating from 20:a6:80:a9:d2:74 by local choice (Reason: 3=DEAUTH_LEAVING)
[ 1484.509285] iwlwifi 0000:02:00.0: L1 Enabled - LTR Disabled
[ 1484.509570] iwlwifi 0000:02:00.0: L1 Enabled - LTR Disabled
[ 1484.509662] iwlwifi 0000:02:00.0: Radio type=0x1-0x3-0x1
[ 1484.733206] iwlwifi 0000:02:00.0: L1 Enabled - LTR Disabled
[ 1484.733450] iwlwifi 0000:02:00.0: L1 Enabled - LTR Disabled
[ 1484.733543] iwlwifi 0000:02:00.0: Radio type=0x1-0x3-0x1
[ 1484.817290] IPv6: ADDRCONF(NETDEV_UP): wlp2s0: link is not ready
[ 1484.858472] iwlwifi 0000:02:00.0: L1 Enabled - LTR Disabled
[ 1484.858816] iwlwifi 0000:02:00.0: L1 Enabled - LTR Disabled
[ 1484.858907] iwlwifi 0000:02:00.0: Radio type=0x1-0x3-0x1
[ 1485.081037] iwlwifi 0000:02:00.0: L1 Enabled - LTR Disabled
[ 1485.081272] iwlwifi 0000:02:00.0: L1 Enabled - LTR Disabled
[ 1485.081379] iwlwifi 0000:02:00.0: Radio type=0x1-0x3-0x1
[ 1485.168117] IPv6: ADDRCONF(NETDEV_UP): wlp2s0: link is not ready
```

Ces logs permettent de tracer le suivi de tous les événements système mais il ne prendront pas en compte les applications ou serveurs utilisateurs (tels **postfix**, **apache**, **nginx**, **prosody**, etc.) qui disposent de leurs propres fichiers de logs.

VI. Droits et permissions

\$ chmod +x

Rendre un fichier exécutable. En fait, lui attribuer des droits en exécution.

Création d'un utilisateur avec les droits **root** :

adduser newUser root

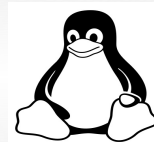
Or

usermod -aG (==ALL GRANTED) root newUser

On peut aussi les ajouter manuellement :

visudo

newUser ALL=(ALL:ALL) ALL



VII. Administration système

II. John

Pour introduire ce chapitre, une fois n'est pas coutume, place à un peu de hacking ! L'auteur de ces lignes a bien conscience que tout le monde n'éprouve pas de plaisir à lire de volumineux pavés techniques (pourtant nécessaires à un bon apprentissage), raison d'être de ce livre.

Aussi, avant de rentrer dans le vif du sujet d'un chapitre important et lourd de nombreux outils essentiels, parfois compliqué, nous avons jugé bon de débiter sur une note de légèreté et de proposer ici une introduction à **john** (nom du paquet linux), outil d'attaque par dictionnaire, plus connu sous le nom de **john the ripper** dans les milieux informés.

Pour cela on apprendra ici la localisation des mots de passe sous **debian**. Il est bien entendu que, davantage qu'un outil de hacking, **john** sert avant tout à tester la fiabilité des mots de passe faibles, qu'il dévoilera.

\$ apt-get install john

A partir de ce chapitre nous ne reviendrons plus sur les notions essentielles abordées précédemment, tel le gestionnaire de paquets APT.

Le mot de passe des utilisateurs sont sauvegardé dans deux fichiers qui sont uniquement accessibles avec des droits **root**.

Le fichier **/etc/passwd** qui contient les noms des comptes utilisateurs (logins) et le fichier **/etc/shadow** qui contient les mots de passes mais cryptés.

On va « hasher » le contenu des 2 fichiers en un seul avec **unshadow** :

```
$ unshadow /etc/passwd /etc/shadow > password.txt
```

On exécute john pour décrypter le fichier crée :

```
$ john password.txt
```

Dans son fonctionnement par défaut **john** va cracker tous les mots de passe compris dans ces fichiers, il est possible de spécifier le nom d'un compte à hacker ou encore son numéro UID avec :

```
$ john --users=nom_utilisateur fichier.extension
```

```
$ john --users=UID_utilisateur fichier.extension
```



```

root@robota:~# unshadow /etc/passwd /etc/shadow > password.txt
root@robota:~# john --users=cracked_user password.txt
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
cracked (cracked user)
lg 0:00:00:00 100% 1/3 1.562g/s 150.0p/s 150.0c/s 150.0C/s cracked_us
er..resu dekarcc
Use the "--show" option to display all of the cracked passwords relia
bly
Session completed
root@robota:~# █

```

On observe dans l'exemple ci-dessus qu'en quelques millisecondes le mot de passe « cracked » est découvert comme ouvrant l'accès à la session « cracked_user ».

Une fois l'exécution de **john** achevée :

\$ john --show password.txt

```

root@robota:~# john --show password.txt
cracked_user:cracked:1002:1003:,,,:/home/cracked_user:/bin/bash

1 password hash cracked, 2 left
root@robota:~# █

```

Cette commande nous donnera les autres mots de passe crackés reliés à la session ciblée. Il apparaît qu'aucun autre mot de passe n'est relié à cet utilisateur. Pour cause, nous venons de le créer. Aussi, apparaît seulement le mot de passe du compte.

Bien sûr le password de l'exemple ne répond pas aux exigences de sécurité et de complexité d'un mot de passe fiable (minimum 10 caractères, majuscules, chiffres, et caractères spéciaux). Le temps de cracking d'un mot de passe par **john** dépend à la fois de ces critères comme de la puissance de la machine exécutante.

Il est du reste possible d'ajouter des options pour spécifier les préférences de caractères brute-forçable par **john** :

```

$ john -incremental:alpha pass.txt (Seulement les lettres)
$ john -incremental:digits pass.txt (Seulement les chiffres)
$ john -incremental:lanman pass.txt (Chiffres, lettres et certains
caractères spéciaux)
$ john -incremental:all pass.txt (Tous les caractères)

```

\$ john --test

Pour connaître la vitesse de cracking de john.

```
root@robota:~# john --test
Benchmarking: descrypt, traditional crypt(3) [DES 128/128 SSE2-16]...
DONE
Many salts:      2903K c/s real, 2903K c/s virtual
Only one salt:   2787K c/s real, 2787K c/s virtual

Benchmarking: bsdicrypt, BSDI crypt(3) ("_J9..", 725 iterations) [DES
128/128 SSE2-16]... DONE
Many salts:      93465 c/s real, 93465 c/s virtual
Only one salt:   91337 c/s real, 91520 c/s virtual

Benchmarking: md5crypt [MD5 32/64 X2]... DONE
Raw:      12822 c/s real, 12848 c/s virtual

Benchmarking: bcrypt ("2a$05", 32 iterations) [Blowfish 32/64 X2]...
DONE
Raw:      700 c/s real, 700 c/s virtual

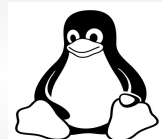
Benchmarking: LM [DES 128/128 SSE2-16]... DONE
Raw:      37909K c/s real, 37909K c/s virtual

Benchmarking: AFS, Kerberos AFS [DES 48/64 4K]... DONE
Short: 365568 c/s real, 366300 c/s virtual
Long:  1158K c/s real, 1158K c/s virtual

Benchmarking: tripcode [DES 128/128 SSE2-16]... DONE
Raw:      2565K c/s real, 2570K c/s virtual

Benchmarking: dummy [N/A]... DONE
Raw:      46152K c/s real, 46244K c/s virtual

Benchmarking: crypt, generic crypt(3) [?/64]... DONE
Many salts:      243763 c/s real, 244251 c/s virtual
Only one salt:   241651 c/s real, 242135 c/s virtual
```



2. Cron :

Cron est un **daemon** permettant d'automatiser les tâches courantes en programmant leur lancement de façon différée. On pense ici notamment aux tâches d'administrations, au lancement de scripts shell pushés sur les postes d'une entreprise par exemple, tel **john**, destiné à vérifier la fiabilité des mots de passe.

2.1 Installation :

```
$ apt-get install cron anacron
```

Installation des paquets **cron** et **anacron** (sur lequel nous reviendrons ultérieurement.)

```
$ /etc/init.d/cron start
```

```
pierre@robota:~$ /etc/init.d/cron start
[ ok ] Starting cron (via systemctl): cron.service.
pierre@robota:~$
```

Ici, on initialise le service servant au lancement du daemon **cron**.

2.2 Crontab :

crontab est le programme utilisé pour installer, désinstaller ou afficher le contenu des tables permettant de piloter le fonctionnement du démon **cron**. Chaque utilisateur dispose de sa propre **crontab**, et bien que celles-ci se trouvent dans **/var/spool/cron/crontabs**, elles ne sont pas conçues pour être modifiées directement.

```
$ crontab -u user -e
```

```
root@robota:~# crontab -u pierre -e
no crontab for pierre - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
```

On crée une **crontab** pour l'utilisateur **pierre** à l'aide de notre éditeur de texte préféré. Dans les exemples qui suivent comme dans tous ceux de cet ouvrage, l'éditeur utilisé sera **vim.basic**.

commande à exécuter.

Par exemple :

```
0 12 1 * 1 john
```

est une instruction qui consiste à dire : Le premier jour du mois, et chaque semaine (1 correspondant au 1^{er} jour de la semaine) à midi, exécute **john** sur le poste.

```
# m h dom mon dow command
0 0 * * * apt-get update && apt-get upgrade
```

Dans cet exemple-ci on demande que chaque jour de chaque mois (donc chaque jour de la semaine aussi) on exécute les commandes nommées à minuit.

On peut commenter d'un # les lignes de la **crontab** que l'on ne souhaite pas voir appliquer immédiatement.

Note : Dans tout langage de programmation (Python, Ruby, C++, etc.) et notamment en langage Shell, un commentaire sur la ligne d'un script permet d'ignorer cette ligne à l'exécution (compilée ou interprétée) et donc de ne pas en tenir compte.

2.3 Droits sur la crontab et l'utilisation de la commande cron :

Si le fichier **/etc/cron.allow** existe, alors vous devez être mentionné (un utilisateur par ligne) dans celui-ci pour pouvoir utiliser cette commande. S'il n'existe pas, mais que le fichier **/etc/cron.deny** existe, alors vous ne devez pas être mentionné dans celui-ci si vous désirez utiliser cette commande.

Si aucun de ces deux fichiers n'existe, alors, selon la configuration du site, soit seul le superutilisateur a le droit d'utiliser cette commande, soit tous les utilisateurs le peuvent.

Si les deux fichiers existent, alors **/etc/cron.allow** sera prioritaire. Cela signifie que **/etc/cron.deny** n'est pas pris en compte et votre identifiant doit être dans **/etc/cron.allow** pour pouvoir utiliser **crontab**.

Indépendamment de l'existence d'un de ces fichiers, le superutilisateur est toujours autorisé à avoir une **crontab**. Sur les systèmes Debian standards, tous les utilisateurs peuvent l'utiliser.

Source :

\$ man crontab



\$ crontab -u pierre -l

Cette commande permet de visualiser la **crontab** courante et applicable de l'utilisateur cible. Voir capture ci-dessous.

```
pierre@robota: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
root@robota:~# crontab -u pierre -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow    command  
# 0 0 * * *          apt-get update && apt-get upgrade  
  
# 0 12 1 * 1         john  
root@robota:~#
```

\$ crontab -u user -r

Suppression de la **crontab** en cours. Remplacer **-r** par l'option **-i** si l'on souhaite demander une confirmation pour la suppression.

3. Anacron :

anacron est sensiblement similaire à **cron**. En fait, les commandes sont les mêmes. A la différence près qu'**anacron** est né pour permettre l'exécution de tâches sans nécessité de laisser la machine allumée 24h/24.

3.1 Principe :

anacron utilise des indications de temps relatives (« une fois par jour / par semaine / par mois ») au lieu de références temporelles absolues (« le 14 janvier 2008 à 15h 30 »). De la sorte, même si vous « manquez » un moment ou une date particulière où l'exécution d'un « job » était prévue, celui-ci sera tout de même exécuté peu de temps après le prochain démarrage du système.

3.2 Utilisation :

Sur un système fonctionnant en permanence, c'est **cron** qui lance **anacron** à 7h30 chaque jour (voir le fichier `/etc/cron.d/anacron`).

```
pierre@robota:~$ cat /etc/cron.d/anacron
# /etc/cron.d/anacron: crontab entries for the anacron package

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

30 7 * * * root test -x /etc/init.d/anacron && /usr/sbin/invo
ke-rc.d anacron start >/dev/null
pierre@robota:~$
```

Sur un système ne tournant pas en permanence, **anacron** est lancé au démarrage car configuré en tant que service.

`$ /etc/init.d/anacron start`

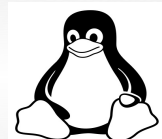
```
pierre@robota:~$ /etc/init.d/anacron start
[ ok ] Starting anacron (via systemctl): anacron.service.
pierre@robota:~$
```

anacron recherche les « timestamps » correspondant à ses différents jobs dans `/var/spool/anacron`.

```
pierre@robota:~$ ls /var/spool/anacron/
cron.daily cron.monthly cron.weekly
pierre@robota:~$
```

Si, d'après un des fichiers timestamps, un job est en attente, **anacron** le lance.

Puis, **anacron** met à jour le fichier timestamp du job en question.



Une fois son travail terminé, anacron s'arrête en attendant la prochaine sollicitation. Il n'y a pas de **daemon anacron** consommant de la mémoire.

Le fonctionnement d'**anacron** est contrôlé par le fichier **/etc/anacrontab** :

```
pierre@robota:~$ cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
HOME=/root
LOGNAME=root

# These replace cron's entries
1      5      cron.daily      run-parts --report /etc/cron.daily
7      10     cron.weekly    run-parts --report /etc/cron.weekly
@monthly 15     cron.monthly   run-parts --report /etc/cron.
monthly

# 1      20     daily_save    rsync -rltgoDv --del --ignore-error
s --force /home/pierre /mnt/point
```

Chaque ligne du fichier correspond à une tâche :

- La 1ère colonne, exprimée en jours indique l'intervalle de temps entre deux exécutions d'une tâche.
- La 2ème colonne, exprimée en minutes, est le délai entre 2 tâches exécutées par anacron.
- La 3ème colonne est le commentaire ajouté qui apparaîtra dans les logs de anacron.
- La 4ème colonne est la tâche à exécuter.

Note : **anacron** n'est pas lancé si le poste utilisé n'est pas sur secteur.

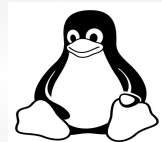


3.3 Quelques options :

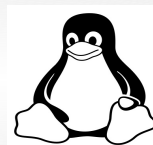
- f : Force l'exécution des tâches, en ignorant les fichiers dateurs.
- u : Met à jour à la date courante les fichiers dateurs des tâches, mais ne lance rien.
- s : Met en série l'exécution des tâches. Anacron ne lancera pas une nouvelle tâche avant que la précédente ne soit terminée.
- n : Lance les tâches tout de suite. Ignore les définitions de délai dans le fichier /etc/anacrontab Cette option implique -s.
- d : N'exécute pas en arrière-plan. Dans ce mode, Anacron enverra les messages d'information vers la sortie d'erreurs, ainsi qu'à syslog. La sortie des tâches est envoyée via un message comme d'habitude.
- q : Supprime les messages destinés à la sortie d'erreurs. Applicable seulement avec -d.
- t **anacrontab** : Utilise le fichier désigné anacrontab, à la place de celui par défaut.
- V : Affiche l'information de version, puis s'arrête.
- h : Affiche un court message d'utilisation, puis s'arrête.

Source :

<https://doc.ubuntu-fr.org/anacron>



4. SSH :



5. FTP:

FTP (File Transfer Protocol) est un protocole Internet de transfert de fichiers.

Il n'est pas natif.

5.1 Installation :

```
# apt-get install ftp ftpd [(daemon)]
```

Le **ftp** est généralement utilisé pour télécharger (*download*) un dossier présent sur un serveur ou au contraire pour envoyer (*upload*) un dossier vers un serveur.

Plusieurs syntaxe sont utilisables pour se connecter via **ftp** :

```
ftp://user@mon-site.domaine
```

```
ftp://user:mot-de-passe@mon-site.domaine/mon-repertoire
```

```
ftp://user:mot-de-passe@mon-site.domaine:port/mon-repertoire
```

Source :

<https://doc.ubuntu-fr.org/ftp>

5.2 Utilisation :

Ces syntaxes sont valables depuis un navigateur, des clients **ftp** comme **FileZilla** ou encore l'explorateur de fichiers **nautilus**. Aussi nous ne nous attarderons pas dessus mais il nous a paru essentiel de signaler ces possibilités.

La syntaxe ftp en ligne de commande est relativement semblable :

```
$ ftp <host-name> <port>
```

Exemples :

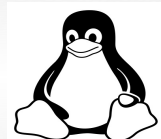
```
$ ftp rtfm.mit.edu
```

```
$ ftp
```

```
ftp > open rtfm.mit.edu
```

Ce serveur est un ancien serveur **ftp** du MIT toujours en activité, originellement destiné à référencer les réponses aux **FAQ**.

Note : Si la réponse à requête **FTP** est « *Name or service not known* », soit l'orthographe du nom est erronée soit votre serveur **FTP** n'est plus



actif.

Le login demandé sur un serveur **FTP** public est « **anonymous** » ou rien.

Le password demandé est rien ou « **anonymous** ».

En dehors de ces codes le serveur auquel vous avez accédé n'est sans doute pas destiné à un usage public.

```
root@robota:~# ftp ftp.wustl.edu
ftp: ftp.wustl.edu: Name or service not known
ftp> bye
root@robota:~# ftp rtfm.mit.edu
Connected to xvm-75.mit.edu.
220 RTFM ftp service
Name (rtfm.mit.edu:pierre): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0      0      3743947 May 28 00:02 Index-byname
-rw-r--r--    1 0      0      499197 May 28 00:02 Index-byname.
gz
-rw-r--r--    1 0      0      3743947 May 28 00:02 Index-bytime
-rw-r--r--    1 0      0      348891 May 28 00:02 Index-bytime.
gz
-rw-r--r--    1 0      1004      0 May 28 00:02 lock
-rw-r--r--    1 0      0      897103 May 28 00:02 ls-lR.Z
drwxrwxr-x   37 0      1004      4096 Jul 11 2002 pub
226 Directory send OK.
ftp> █
```

A partir de cet instant la connexion est établie (code 230) et vous pouvez naviguez dans son arborescence. Cependant n'utilisez pas la complétion automatique (**TAB**) en console car c'est alors l'arborescence de votre session et vos commandes qui vont apparaître. En effet même si vous avez initiée une connexion depuis votre console vous demeurez connecté en tant que vous-même.

Exemple :



```
ftp>
.ICEauthority          .selected_editor
.Xauthority            .ssh/
.bash_history          .vim/
.bashrc               .viminfo
.cache/               .wget-hsts
.config/              .xsession-errors
```

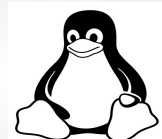
Ainsi lorsque vous déplacerez dans l'arborescence du serveur pensez à écrire les commandes complètes et à vérifier le répertoire courant par \$ pwd.

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0      0          3743947 May 28 00:02 Index-byname
-rw-r--r--    1 0      0          499197 May 28 00:02 Index-byname.
gz
-rw-r--r--    1 0      0          3743947 May 28 00:02 Index-bytime
-rw-r--r--    1 0      0          348891 May 28 00:02 Index-bytime.
gz
-rw-r--r--    1 0      1004        0 May 28 00:02 lock
-rw-r--r--    1 0      0          897103 May 28 00:02 ls-lR.Z
drwxrwxr-x   37 0      1004        4096 Jul 11 2002 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> pwd
257 "/pub"
ftp> █
```

Vous serez averti d'un changement de répertoire par le code de retour **FTP 250**.

Ce code pouvant avoir de multiples significations, renvoie les messages générés par le serveur.

5.3 Download and upload en ftp :



6. Les variables du systeme :

\$ locale

Affiche les informations sur la localisation régionale.

```
pierre@robota:~$ locale
LANG=fr_FR.UTF-8
LANGUAGE=
LC_CTYPE="fr_FR.UTF-8"
LC_NUMERIC="fr_FR.UTF-8"
LC_TIME="fr_FR.UTF-8"
LC_COLLATE="fr_FR.UTF-8"
LC_MONETARY="fr_FR.UTF-8"
LC_MESSAGES="fr_FR.UTF-8"
LC_PAPER="fr_FR.UTF-8"
LC_NAME="fr_FR.UTF-8"
LC_ADDRESS="fr_FR.UTF-8"
LC_TELEPHONE="fr_FR.UTF-8"
LC_MEASUREMENT="fr_FR.UTF-8"
LC_IDENTIFICATION="fr_FR.UTF-8"
LC_ALL=
```

locale-gen « en_US.UTF-8 »

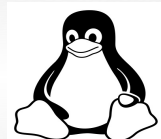
Redéfinit les paramètres régionaux sur la langue ciblée.

dpkg-reconfigure locales

Reconfigure via **dpkg** les paramètres régionaux renseignés par la commande **locale-gen**. Generating locales.

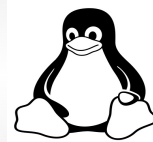
dpkg-reconfigure tzdata data

Change les paramètres UTC/GMT en lieu et place de « data ».

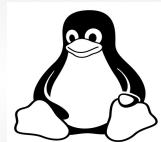


VIII. Initiation à Metasploit framework

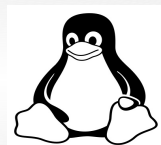
L'installation de **metasploit framework** (aussi nommé **MSF-console**) sous Debian nécessiterait à elle seule un chapitre entier. Aussi l'auteur de ces lignes suppose que vous avez déjà installé **metasploit** ou que vous disposez d'un os live orienté sécurité tel l'excellent **Kali Linux** dont le **kernel** est « based on Debian ».



IX. Initiation à la Programmation shell



X. Le Kernel (systemd)



XI. Services réseaux

1. Quelques commandes :

1.1 Wget :

La commande **wget** permet de récupérer par **download** un fichier ciblé défini par une **url**.

La syntaxe est la suivante :

```
$ wget http://server/file_to_download.ext
```

Cependant, la richesse de **wget** est vaste et l'on peut être amené à download un fichier texte tout comme un fichier **.iso** par exemple, beaucoup plus lourd.

Aussi, pour continuer à travailler en réseau sans être impacté par le téléchargement on pourra être tenté de limiter la vitesse de téléchargement de **wget** afin de ne pas saturer la bande passante.

```
$ wget --limit-rate 256k http://server/file.iso
```

L'ajout de l'option **--limit-rate** permet de limiter la vitesse de téléchargement à la vitesse souhaitée, dans les limites de votre débit réseau.

On peut préciser la vitesse en **kilobytes (k)** ou en **megabytes (m)**.

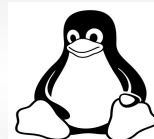
1.2 Sendmail :

Il est possible d'envoyer des mails directement depuis le terminal ou par le biais de script en utilisant la commande **sendmail**.

Pour son bon fonctionnement il faudra au préalable, acquérir les paquets **libio-socket-ssl-perl**, **libnet-ssleay-perl** et **sendmail**.

Syntaxe et exemple d'utilisation :

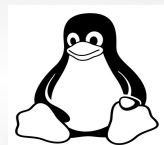
```
$ sendmail -f from_user@gmail.com \  
-t to_user@mail_domain.com \  
-u « objet du mail » -m « message a envoyer » \  
-s smtp.gmail.com \  
-o tls=yes -xu gmail_account \  
-xp gmail_password
```



2. Administration serveur :



XII. Hacking and forensic



XIII. Quelques exemples d'utilisation de commandes avancées



XIV. Contributeurs et remerciements

