

La commande `ls`

Pour connaître plus sur la commande `ls`, on utilise la commande `man` qui nous donne le manuel d'une commande.

```
theo@debiantheo:~$ man ls
```

```
LS(1)                                Commandes de l'utilisateur                                LS(1)

NOM
ls - Afficher le contenu de répertoires

SYNOPSIS
ls [OPTION]... [FICHIER]...

DESCRIPTION
Afficher les informations des FICHIERS (du répertoire courant par défaut). Les entrées sont triées alphabétiquement si aucune des options -cftuvSUX ou --sort n'est indiquée.

Les paramètres obligatoires pour les options de forme longue le sont aussi pour les options de forme courte.

-a, --all
    inclure les entrées débutant par « . »

-A, --almost-all
    omettre les fichiers « . » et « .. »

--author
    Manual page ls(1) line 1 (press h for help or q to quit)
```

Essentiellement, `ls` est une commande qui permet d'afficher le contenu du répertoire où nous nous situons, cependant cette commande par elle-même n'est pas très détaillée, dans ce cas là, il faut utiliser des *options*.

```
theo@debiantheo:~$ ls
Bureau Documents Images Modèles Musique Public Téléchargements Vidéos
theo@debiantheo:~$
```

Les options permettent de spécifier ce qu'on veut faire avec une commande, dans ce cas-là, on voudrait voir les fichiers et dossiers cachés, et aussi avoir une vue plus détaillée qui nous permet de voir des permissions.

Il y a deux syntaxes principales pour les options, en forme longue et en abrégé:

```
theo@debiantheo:~$ ls --all
.          .config      Images      Public
..         .dmrc         .lesshtst  .sudo_as_admin_successful
.bash_history Documents    .local     Téléchargements
.bash_logout .face       Modèles    Vidéos
.bashrc     .face.icon  .mozilla   .viminfo
Bureau     .gnupg      Musique    .Xauthority
.cache     .ICEauthority .profile   .xsession-errors

theo@debiantheo:~$ ls -a
.          .config      Images      Public
..         .dmrc         .lesshtst  .sudo_as_admin_successful
.bash_history Documents    .local     Téléchargements
.bash_logout .face       Modèles    Vidéos
.bashrc     .face.icon  .mozilla   .viminfo
Bureau     .gnupg      Musique    .Xauthority
.cache     .ICEauthority .profile   .xsession-errors
theo@debiantheo:~$
```

```
theo@debiantheo:~$ ls -l
total 32
drwxr-xr-x 2 theo theo 4096 27 sept. 11:14 Bureau
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Documents
drwxr-xr-x 2 theo theo 4096 27 sept. 11:42 Images
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Modèles
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Musique
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Public
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Téléchargements
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Vidéos
theo@debiantheo:~$
```

A gauche, l'option pour voir les fichiers cachés, et à droite l'option pour avoir une vue détaillée.

```
theo@debiantheo:~$ ls -la
total 108
drwx----- 15 theo theo 4096 27 sept. 11:42 .
drwxr-xr-x  3 root root 4096 27 sept. 10:50 ..
-rw-----  1 theo theo  384 27 sept. 11:27 .bash_history
-rw-r--r--  1 theo theo  220 27 sept. 10:50 .bash_logout
-rw-r--r--  1 theo theo 3526 27 sept. 10:50 .bashrc
drwxr-xr-x  2 theo theo 4096 27 sept. 11:14 Bureau
drwxr-xr-x  8 theo theo 4096 27 sept. 11:11 .cache
drwx-----  7 theo theo 4096 27 sept. 11:02 .config
-rw-r--r--  1 theo theo   35 27 sept. 10:58 .dmrc
drwxr-xr-x  2 theo theo 4096 27 sept. 10:58 Documents
```

Pour utiliser les deux options ensemble

Les commandes `cat`, `head` et `tail`

Afin d'afficher le contenu d'un fichier entièrement, il faut utiliser la commande `cat`, qui imprime le contenu du fichier visé dans le shell, par exemple le fichier `~/ .bashrc`, un fichier de configuration pour le shell Bash.

Pour afficher les premières et dernières lignes d'un fichier, il faut utiliser les commandes `head` et `tail` respectivement, cependant il faut spécifier le nombre de lignes qu'on veut afficher avec l'option `-n [nombre]`

```
theo@debiantheo:~$ head -n 10 .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

theo@debiantheo:~$
```

```
theo@debiantheo:~$ cat .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
```

Pour afficher 10 lignes.

```
theo@debiantheo:~$ head -n 20 .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
theo@debiantheo:~$
```

```
theo@debiantheo:~$ tail -n 10 .bashrc
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
theo@debiantheo:~$
```

```
theo@debiantheo:~$ tail -n 20 .bashrc
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
theo@debiantheo:~$
```

Pour afficher 20 lignes

`apt`, le gestionnaire de paquet de Debian, et `wget`

`apt` est une commande essentielle pour gérer ses logiciels sur Debian.

Par exemple, pour installer `cmatrix`, il faut faire `sudo apt install cmatrix`

Ensuite on exécute `cmatrix`

```
theo@debiantheo:~$ sudo apt install cmatrix
[sudo] Mot de passe de theo :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Paquets suggérés :
  cmatrix-xfont
Les NOUVEAUX paquets suivants seront installés :
  cmatrix
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 17,5 ko dans les archives.
Après cette opération, 53,2 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 cmatrix amd64 2
.0-3 [17,5 kB]
17,5 ko réceptionnés en 0s (224 ko/s)
Sélection du paquet cmatrix précédemment désélectionné.
(Lecture de la base de données... 118774 fichiers et répertoires déjà installés.
)
Préparation du dépaquetage de .../cmatrix_2.0-3_amd64.deb ...
Dépaquetage de cmatrix (2.0-3) ...
Paramétrage de cmatrix (2.0-3) ...
Traitement des actions différées (« triggers ») pour mailcap (3.70+nmul) ...
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.26-1)
...
Q / O f X Q _ 7 E > @ ^ R + D L
M r 7 # w G p + 0 S q \ s + j
D F l 4 E P j p _ 7 0 H % t
P N # \ d 0 ^ 7 t l ^ 1 0 J 8 n
q ~ j h ~ 5 6 ^ p K S t d
l U o v ~ h _ b ^ H L w l 3 q
( t ~ 1 0 c S h H j 9 6
h ~ 5 9 h ^ F 2 / q 1 6
~ \ 0 C 0 1 H 0 > 0 j 0 3
0 0 j v 0 1 # _ j ^ n + 0
0 < + 1 j 0 7 7 W Q 2 \ W n j f
g 0 0 0 1 + + 1 l n ^ z q 4 w
1 1 7 F H k s F k ^ 7 0 c
0 0 0 4 0 1 9 3 1 ^ % 0 j d 4
0 + 0 1 1 4 l 1 2 ^ 0 ^ R l
2 k 0 6 + 0 + 6 7 7 V ^ P ^ F l
0 W R ^ 5 0 1 1 1 0 _ 0 y ^ e j ^ 0 j
% 0 0 + 1 3 s 3 2 9 j 4 2 1 ^ 0 0 o l 0 t u
> j A ^ A 3 F V h ^ 5 _ j M l 7 j f ^ - & k 2
1 1 j M - j H 2 1 D W p L j ^ 0 6 j l f s
v ! ^ V Y ^ j R 0 E - C u V F j w ^ F 0 0 " ^
M E Y ^ 2 ^ & H K 9 F u 6 T Y 2 1 x D 9 j U
T + 6 4 : l U 0 - 2 H u P y 2 ^ V ^ A & u V
C ^ k l t q ^ Y F E / a B w E + x 7 e f s l
```

C'est un programme très utile pour faire semblant d'être un hacker :)

Une mise à jour se fait en 2 étapes.

En premier, il faut mettre à jour les informations, fournies par les sources, sur les paquets qui sont dans le dépôt Debian. On fait ça avec `sudo apt update`

Ensuite, on met à jour les paquets qui sont installés dans notre système avec

`sudo apt upgrade`

```
theo@debiantheo:~$ sudo apt update
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Réception de :2 http://security.debian.org/debian-security bookworm-security InR
elease [48,0 kB]
Réception de :3 http://deb.debian.org/debian bookworm-updates InRelease [52,1 kB
]
100 ko réceptionnés en 1s (91,7 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
theo@debiantheo:~$
```

```
theo@debiantheo:~$ sudo apt upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
theo@debiantheo:~$
```

Si nous voulons télécharger quelque chose sur internet, on utilise la commande `wget`, par exemple, on peut télécharger la page `google.com`.

```
theo@debiantheo:~$ wget google.com
--2023-09-28 10:09:34-- http://google.com/
Résolution de google.com (google.com)... 142.251.37.238, 2a00:1450:4006:812::200e
Connexion à google.com (google.com)|142.251.37.238|:80... connecté.
requête HTTP transmise, en attente de la réponse... 301 Moved Permanently
Emplacement : http://www.google.com/ [suivant]
--2023-09-28 10:09:35-- http://www.google.com/
Résolution de www.google.com (www.google.com)... 142.250.201.4, 2a00:1450:4006:803::2004
Connexion à www.google.com (www.google.com)|142.250.201.4|:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : non indiqué [text/html]
Sauvegarde en : « index.html »

index.html          [ <=>          ]  22,29K  --.-KB/s    ds 0,02s

2023-09-28 10:09:35 (1,11 MB/s) - « index.html » sauvegardé [22827]
```

```
theo@debiantheo:~$ ls
Bureau      Images      Modèles    Public      Vidéos
Documents  index.html  Musique    Téléchargements
theo@debiantheo:~$
```

Maintenant, il faut qu'on redémarre ou éteint la machine, les commandes sont, respectivement, `sudo reboot` et `sudo shutdown now` (on met `now` sinon on doit attendre).

Création et gestion de comptes et de groupes

```
theo@debiantheo:~$ nano users.txt
```

```
GNU nano 7.2 users.txt *
User1
User2
```

Avant tout, nous allons créer un fichier `users.txt` avec `nano`, un éditeur de texte basique, nous allons écrire `User1` et `User2`.

```
[ Nouveau fichier ]
^G Aide      ^O Écrire    ^N Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich ^M Remplacer ^U Coller    ^J Justifier ^V Aller ligne
```

En premier, nous allons créer un nouveau groupe qui s'appelle `Plateformeurs` avec la commande `groupadd [...]`. Ensuite, on crée deux utilisateurs, `User1` et `User2` avec l'aide de la commande `useradd [...]`. Puis nous ajoutons `User2` au groupe `Plateformeurs` avec la commande `usermod -aG [groupe] [utilisateur]`, l'option `-aG` veut dire "ajouter à un Groupe".

```
theo@debiantheo:~$ sudo groupadd Plateformeurs
theo@debiantheo:~$ sudo useradd User1
theo@debiantheo:~$ sudo useradd User2
theo@debiantheo:~$ sudo usermod -aG Plateformeurs User2
theo@debiantheo:~$
```

Maintenant, nous allons gérer les droits des utilisateurs. Mais avant tout, il

faut aussi comprendre comment lire le résultat `ls -l`

```
theo@debiantheo:~$ ls -l
total 56
drwxr-xr-x 2 theo theo 4096 27 sept. 13:15 Bureau
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Documents
drwxr-xr-x 2 theo theo 4096 28 sept. 10:29 Images
-rw-r--r-- 1 theo theo 22827 28 sept. 10:09 index.html
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Modèles
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Musique
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Public
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Téléchargements
drwxr-xr-x 2 theo theo 4096 27 sept. 10:58 Vidéos
theo@debiantheo:~$
```

Le premier caractère désigne le type de fichier. Les trois prochains caractères représente les droits du

propriétaire, ensuite il y a ceux des membres du groupe, après les permissions des autres utilisateurs.

Ensuite il y a le nombre de liens direct pour accéder à ce fichier, suivi du nom du propriétaire et le groupe, puis la taille du fichier, la date de la dernière modification, et enfin le nom du fichier.

Pour les permissions, "r" veut dire lecture, "w" veut dire écriture et "x" veut dire exécution.

Premièrement, on va copier le fichier 'users.txt' deux fois, une fois en le renommant 'droits.txt' et l'autre fois en 'groupes.txt' avec 'cp'.

```
theo@debiantheo:~$ cp users.txt droits.txt
theo@debiantheo:~$ cp users.txt groupes.txt
theo@debiantheo:~$
```

Pour les prochaines étapes, il faudra utiliser 'sudo'

Deuxièmement, on va changer le propriétaire de 'droits.txt' avec 'chown' pour que ça soit "User1".

```
theo@debiantheo:~$ sudo chown User1 droits.txt
theo@debiantheo:~$ ls -l droits.txt
-rw-r--r-- 1 User1 theo 12 28 sept. 11:25 droits.txt
theo@debiantheo:~$
```

Troisièmement, avec 'chmod' on va changer les droits du fichier 'droits.txt' afin que "User2" ne peut que lire le fichier, puis on va changer les droits de 'groupes.txt' pour que tous les utilisateurs ne peuvent que le lire.

```
theo@debiantheo:~$ sudo chmod o=r droits.txt
theo@debiantheo:~$ ls -l droits.txt
-rw-r--r-- 1 User1 theo 12 28 sept. 11:25 droits.txt
theo@debiantheo:~$
```

"o" désigne les utilisateurs qui ne sont ni le propriétaire, ni membre du groupe.

```
theo@debiantheo:~$ sudo chmod o=r groupes.txt
theo@debiantheo:~$ ls -l groupes.txt
-rw-r--r-- 1 theo theo 12 28 sept. 11:26 groupes.txt
theo@debiantheo:~$
```

Quatrièmement, on va désigner "Plateformeurs" comme groupe du fichier avec 'chown' ensuite on va changer les droits du fichier avec 'chmod' afin que les membres du groupe "Plateformeurs" puissent lire et écrire dans 'groupes.txt'

```
theo@debiantheo:~$ sudo chown theo:Plateformeurs groupes.txt
theo@debiantheo:~$ sudo chmod g=wr groupes.txt
theo@debiantheo:~$ ls -l groupes.txt
-rw-rw-r-- 1 theo Plateformeurs 12 28 sept. 11:26 groupes.txt
theo@debiantheo:~$
```

"g" désigne les membres du groupe

.bashrc, les alias et les variables d'environnement

'~/ .bashrc' est un `script` qui est exécuté lorsqu'un utilisateur se connecte, ça permet à un utilisateur de pouvoir configurer son shell de façon plus optimale pour lui.

L'un des types de configuration les plus utilisés sont les `alias`, qui est une commande qui permet de lancer une autre commande qui est typiquement plus complexe.

Par exemple, pour créer un alias `la` qui nous permettrait d'exécuter `ls -la` sans avoir à taper les options, il faut aller dans le `~/.bashrc`.

Allons au bas de la page dans `~/.bashrc`, et mettons un `commentaire` avec `"#"` pour faire un titre, de façon à ce que ça soit facile à repérer.

```
# ALIAS CUSTOM
alias la='ls -la'
```

il faut faire `alias [nom de l'alias]='command'`

Nous allons maintenant créer deux alias pour raccourcir `sudo apt update` et `sudo apt upgrade` en `update` et `upgrade`.

```
# ALIAS CUSTOM
alias la='ls -la'
alias update='sudo apt update'
alias upgrade='sudo apt upgrade'
```

Désormais, on va créer une `variable d'environnement` qui est égale au nom d'utilisateur.

```
# VARIABLES CUSTOM
export USER=$(whoami)
```

`whoami` est une commande qui donne le nom de l'utilisateur qui l'exécute.

Afin que les modifications qu'on a fait dans le shell s'appliquent, on doit faire:

```
source ~/.bashrc
```

```
theo@debiantheo:~$ source ~/.bashrc
theo@debiantheo:~$ la
total 152
drwxrwxr-x 15 theo theo 4096 28 sept. 13:43 .
drwxr-xr-x  3 root root 4096 28 sept. 10:26 ..
-rw-r----- 1 theo theo 5359 28 sept. 12:18 .bash_history
-rw-r--r--  1 theo theo 220 27 sept. 10:50 .bash_logout
-rw-r--r--  1 theo theo 3666 28 sept. 13:43 .bashrc
drwxr-xr-x  2 theo theo 4096 28 sept. 12:18 Bureau
drwxr-xr-x  8 theo theo 4096 27 sept. 11:11 .cache
drwxr----- 7 theo theo 4096 27 sept. 11:02 .config
-rw-r--r--  1 theo theo 35 27 sept. 13:37 .dmrc
drwxr-xr-x  2 theo theo 4096 27 sept. 10:58 Documents
-rw-r--r--  1 User1 theo 12 28 sept. 11:25 droits.txt
-rw-r--r--  1 theo theo 5290 27 sept. 10:50 .face
lrwxrwxrwx  1 theo theo 5 27 sept. 10:50 .face.icon -> .face
drwxr----- 3 theo theo 4096 27 sept. 10:58 .gnupg
-rw-rw-r--  1 theo Plateformeurs 12 28 sept. 11:26 groupes.txt
-rw-r----- 1 theo theo 0 27 sept. 10:58 .ICEauthority
drwxr-xr-x  2 theo theo 4096 28 sept. 13:43 Images
-rw-r--r--  1 theo theo 22827 28 sept. 10:09 index.html
-rw-r----- 1 theo theo 57 28 sept. 13:38 .lessht
drwxr----- 3 theo theo 4096 27 sept. 10:58 .local
drwxr-xr-x  2 theo theo 4096 27 sept. 10:58 Modèles
```

```
theo@debiantheo:~$ update
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
e
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
theo@debiantheo:~$ upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
theo@debiantheo:~$ echo $USER
theo
theo@debiantheo:~$
```


Afin de lister toutes les variables d'environnement, on utilise la commande ``set``, on peut voir la variable `$USER` qu'on a créé, et aussi on voit la variable `$PATH` sur laquelle on va se concentrer, il faudrait peut-être plutôt faire ``set | less`` au cas où on ne peut pas défiler jusqu'en haut.

```
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
PIPESTATUS=( [0]="0" [1]="127" )
PPID=1127
PS1='\[\e]0;\u@h: \w\a\}${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
PS2='> '
PS4='+ '
PWD=/home/theo
QT_ACCESSIBILITY=1
SESSION_MANAGER=local/debiantheo:@/tmp/.ICE-unix/709,unix/debiantheo:/tmp/.ICE-unix/709
SHELL=/bin/bash
```

Cette variable pointe vers là où les `binaires`, nos commandes bash, sont situés, on souhaite ajouter ``~/Bureau``.

```
# VARIABLES CUSTOM
export USER=$(whoami)
export PATH=$PATH:~/Bureau
```

On **ajoute** le répertoire à la fin de `$PATH`, si on fait ``PATH=~/Bureau``, on ne pourra plus utiliser nos commandes, à part celles qui sont dans ``/sbin``

```
theo@debiantheo:~$ source ~/.bashrc
theo@debiantheo:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/theo/Bureau
theo@debiantheo:~$
```

``tar``, est comment extraire les archives .tar

Après avoir téléchargé un fichier `.tar`, on utilise ``tar -xf``

```
theo@debiantheo:~/Téléchargements$ tar -xf Ghost\ in\ the\ Shell.tar
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.quarantine » pour l'en-tête étendu a été ignoré
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.provenance » pour l'en-tête étendu a été ignoré
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.macl » pour l'en-tête étendu a été ignoré
theo@debiantheo:~/Téléchargements$ ls
'Ghost in the Shell.pdf'  'Ghost in the Shell.tar'
theo@debiantheo:~/Téléchargements$
```

Les messages d'avertissement sont causés car le `.tar` a été créé sur un Mac, rien de grave. L'option ``-x`` veut dire extraire, et l'option ``-f`` est pour désigner un fichier.

Utilisation des redirections dans Bash

Il y a 4 types de redirections qui sont utilisés dans Bash:

- `<`, qui permet de prendre les données dans un fichier pour les donner à une commande, cependant ça peut être parfois redondant

```
theo@debiantheo:~$ cat lettres      theo@debiantheo:~$ sort < lettres  theo@debiantheo:~$ sort lettres
p                                     a                                     a
m                                     b                                     b
q                                     e                                     e
g                                     g                                     g
v                                     m                                     m
u                                     p                                     p
b                                     q                                     q
r                                     r                                     r
e                                     u                                     u
a                                     v                                     v
```

`sort` permet de ranger une liste en ordre alphabétique.

- `>`, qui permet de prendre l'output d'une commande et la mettre dans un fichier.

```
theo@debiantheo:~$ echo $PATH > inredirect
theo@debiantheo:~$ cat inredirect
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/theo/Bureau
```

- `>>`, qui permet de prendre l'output d'une commande et de **l'ajouter** à un fichier.

```
theo@debiantheo:~$ echo $USER >> inredirect
theo@debiantheo:~$ cat inredirect
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/theo/Bureau
theo
```

- `|`, qui permet de prendre l'output d'une commande et de la passer à une autre commande.

```
theo@debiantheo:~$ cat .bashrc | lolcat
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend
```

- `&&` n'est pas une redirection, mais elle permet d'enchaîner à la prochaine commande si la précédente a réussi.

Avec ces redirections, nous allons exécuter une chaîne de commandes plus complexes et spécifiques, en une seule ligne.

Nous allons:

- Créer un fichier `'une_commande.txt'` et écrire dedans avec ``echo``, on insère le résultat de ``echo`` avec ``>``
- Nous allons compter les nombres de lignes dans `'/etc/apt/sources.list'` avec ``wc -l`` (l'option ``-l`` permet de spécifier qu'on veut compter des lignes), puis on met ce résultat dans le fichier `'nb_lignes.txt'`
- On affiche le contenu de `'/etc/apt/sources.list'` avec ``cat``
- On utilise ``cat`` avec `'/etc/apt/sources'` et on met le résultat dans le fichier `'save_sources'`
- On va utiliser ``find``, pour rechercher des fichiers et des données, et ``grep``, pour rechercher dans les fichiers, afin de trouver tous les fichiers commençant par ``.`` et qui contiennent `"alias"`

```
theo@debiantheo:~$ echo Je suis un fichier texte > une_commande.txt && wc -l /etc/apt/sources.list > nb_lignes.txt && cat /etc/apt/sources.list && cat /etc/apt/sources.list > save_sources && find -type f -name '.*' -exec grep -H 'alias' {} \;
```

```
#deb cdrom:[Debian GNU/Linux 12.1.0 _Bookworm_ - Official amd64 NETINST with firmware 20230722-10:48]/bookworm main non-free-firmware
```

```
deb http://deb.debian.org/debian/ bookworm main non-free-firmware
deb-src http://deb.debian.org/debian/ bookworm main non-free-firmware
```

Voici les commandes, et l'affichage des sources.

```
./.bashrc:#alias l='ls -CF'
./.bashrc:# ~/.bash_aliases, instead of adding them here directly.
./.bashrc:if [ -f ~/.bash_aliases ]; then
./.bashrc:    . ~/.bash_aliases
./.bashrc:alias la='ls -la'
./.bashrc:alias update='sudo apt update'
./.bashrc:alias upgrade='sudo apt upgrade'
./.bash_history:grep .bashrc alias
./.bash_history:grep .bashrc *alias*
```

Voici les fichiers, qui contiennent qui commencent avec ``.`` et qui contiennent `"alias"`.

```
theo@debiantheo:~$ cat une_commande.txt
Je suis un fichier texte
theo@debiantheo:~$ cat nb_lignes.txt
18 /etc/apt/sources.list
theo@debiantheo:~$ cat save_sources
#deb cdrom:[Debian GNU/Linux 12.1.0 _Bookworm_ - Official amd64 NETINST with firmware 20230722-10:48]/bookworm main non-free-firmware
```

```
deb http://deb.debian.org/debian/ bookworm main non-free-firmware
deb-src http://deb.debian.org/debian/ bookworm main non-free-firmware
```

Confirmation que nos commandes d'écriture de fichier ont marché.