

Documentation

Shell

Par Anaïs GILG

22 SEPTEMBRE 2022

Sommaire

Qu'est-ce qu'un Shell ?

Les différents Shell qu'il existe

Introduction aux Jobs

JOB1

JOB2

MacOS

JOB3

JOB4

JOB5

JOB6

JOB7

Aller plus loin...

Linux

JOB3

JOB4

JOB5

JOB6

JOB7

Aller plus loin...

Qu'est-ce qu'un Shell ?

Un Shell aussi appelé Interface Système ou coquille, interprète des lignes de commande envoyé par l'utilisateur depuis son clavier vers son terminal.

Il nous permet d'accéder à notre OS et d'interagir directement avec lui.

Il peut se présenter sous deux formes :

- Soit une interface en ligne de commande ou CLI pour "Command Line Interface"
- Soit une interface graphique ou GUI pour "Graphical User Interface"

L'interface graphique est composée d'un écran d'accueil, d'un bureau, d'icônes pour représenter les fichiers et les dossiers et de champs de recherche, de filtres, etc.

Alors que l'interface en ligne de commande se fait sans l'usage de la souris, seulement avec le clavier. Il faut donc connaître certaines commandes et leur syntaxe.

Les différents Shell qu'il existe

Il existe différents Shell, voici les plus connus :

- sh (Bourne Shell)
- ksh (Korn Shell)
- csh (cShell)
- bash (Bourn Again Shell)
- zsh (zShell)

Pour savoir quel Shell on utilise il est possible d'utiliser la commande suivante dans son terminal :

`ps -p $$`

Ou bien :

`echo $SHELL`

Pour consulter la liste des Shells déjà installés sur notre OS, il faut utiliser :

`cat /etc/shells`

Introduction aux Jobs

Dans cette documentation nous allons voir aux files des Jobs les différences entre MacOS et Linux. Nous verrons qu'en plus de l'esthétique de LIC, il y a aussi certaines commandes qui changent.

Avec MacOS nous utiliserons le Shell zsh alors d'avec Linux se sera bash.

A savoir zsh est une version améliorer de bash et peut aussi être utiliser sur Linux et vice versa pour bash qui peut être utiliser sur MacOS.

Pour cela il faudra écrire dans le terminal :

`chsh -s /bin/leshellssouhaité`
exemple : `chsh -s /bin/bash`

Dans les deux premiers Jobs les commandes sont les mêmes.

JOB1

Pour afficher le manuel d'une commande il faut utiliser : `man`

Donc pour voir le manuel de ls il suffit de faire : `man ls`

Pour afficher les fichiers cachés on utilise : `ls -a`

La commande `ls` permet de lister le contenu de l'endroit où nous nous trouvons.

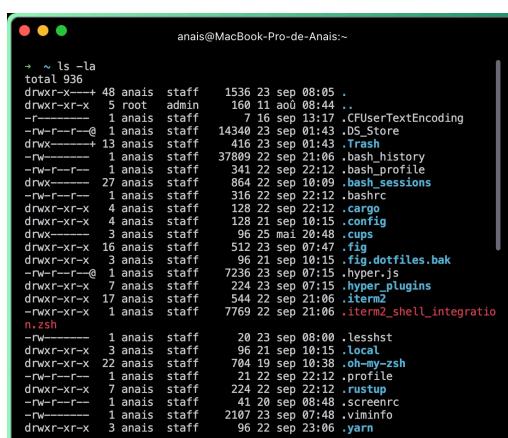
L'option `-a` permet de montrer les fichiers qui sont cachés.

Pour afficher les fichiers cachés plus les infos sur les droits sous fourmes de liste : `ls -la`

`ls -l` permet de lister le contenu de l'endroit où nous nous trouvons plus les détails sur de chaque fichiers et dossiers qui s'y trouve.

Nous pouvons combiner c'est deux options ensemble ce qui donnera : `ls -la`
(Comme chacune de ces deux option sont constituer d'une lettre on peut les mettre ensemble après le tiret.)

Voici ce qu'y s'affiche dans le terminal quand nous utilisons : ls -la



```
anaïs@MacBook-Pro-de-Anaïs:~  
+ ~ ls -la  
total 936  
drwxr-x---+ 48 anais staff 1536 23 sep 08:05 .  
drwxr-xr-x  5 root  admin 160 11 aoû 08:44 ..  
-r-----  1 anais staff   7 16 sep 13:17 .CFUserTextEncoding  
-rw-r--r--@ 1 anais staff 14340 23 sep 01:43 .DS_Store  
drwxr-x---+ 13 anais staff 416 23 sep 01:43 .Trash  
-r-----  1 anais staff 37889 22 sep 21:08 .bash_history  
-rw-r--r--@ 1 anais staff 349 22 sep 22:10 .bash_profile  
drwxr-x--- 27 anais staff 604 22 sep 22:49 .bash_sessions  
drwxr--r--  1 anais staff 316 22 sep 22:12 .cache  
drwxr-xr-x  4 anais staff 128 22 sep 22:12 .cargo  
drwxr-xr-x  4 anais staff 128 21 sep 10:15 .config  
drwxr-x---  3 anais staff  96 25 mai 20:48 .cups  
drwxr-xr-x 16 anais staff 512 23 sep 07:47 .fig  
drwxr-xr-x  3 anais staff  96 21 sep 10:15 .fig_dotfiles.bak  
-rw-r--r--@ 1 anais staff 723 23 sep 07:15 .hyper.js  
drwxr-xr-x  7 anais staff 224 23 sep 07:15 .hyper_plugins  
drwxr-xr-x 17 anais staff 544 22 sep 21:05 .item2  
-rwxr-xr-x  1 anais staff 7769 22 sep 21:05 .item2_shell_integratio  
n.zshrc  
-r-----  1 anais staff 20 23 sep 08:00 .lesshst  
drwxr-xr-x  3 anais staff 96 21 sep 10:15 .local  
drwxr-xr-x 22 anais staff 784 19 sep 10:38 .oh-my-zsh  
-rwxr--r--@ 1 anais staff 21 22 sep 22:12 .profile  
drwxr-xr-x  7 anais staff 224 22 sep 22:12 .rustup  
-rwxr--r--@ 1 anais staff 41 20 sep 08:48 .screenrc  
-rwxr----- 1 anais staff 2107 23 sep 07:48 .viminfo  
drwxr-xr-x  3 anais staff 96 22 sep 23:06 .yarn
```

Question 1 :

Chaque commande offre des options qui peuvent modifier son comportement. Les options sont souvent nommées par une seule lettre et précédées d'un tiret « - »

Question 2 :

Les deux syntaxes principales pour écrire des options de commande sont :

1. L'élément + « - » + l'option ou les options contenant une lettre chacune
2. L'élément + « - - » + l'option qui celle-ci contient plusieurs lettres.

JOB2

Pour ce Job il y a seulement le nom du fichier qui change pour bash le fichier se nomme : .bashrc et pour zsh se sera : .zshrc

Pour afficher les 10 premières lignes d'un fichier c'est avec la commande : **head** qui va automatiquement afficher les 10 premières lignes.

Et pour afficher les 20 premières lignes (ou bien le nombre souhaité en l'occurrence ici c'est 20) il faut ajouter une option « **-n** + le nombre souhaité » à l'élément **head**

Ce qui donne donc pour les 10 premières lignes :

head + le nom du fichier
head .zshrc

Ce qui donne donc pour les 20 premières lignes :

head + -n20 + le nom du fichier
head -n20 .zshrc

Pour afficher les 10 dernières lignes d'un fichier c'est avec la commande : **tail**

qui va automatiquement afficher les 10 premières lignes.

Et pour afficher les 20 premières lignes (ou bien le nombre souhaité en l'occurrence ici c'est 20) il faut ajouter une option « **-n** » + le nombre souhaité » à l'élément **tail**

Ce qui donne donc pour les 10 premières lignes :

head + le nom du fichier
head .zshrc

Ce qui donne donc pour les 20 premières lignes :

tail + -n20 + le nom du fichier
tail -n20 .zshrc

```
anais@MacBook-Pro-de-Anais:~
+ ~ head .zshrc
# Fig pre block. Keep at the top of this file.
[[ -f "$HOME/.fig/shell/zshrc.pre.zsh" ]] && builtin source "$HOME/.fig/shell/zshrc.pre.zsh"
# export PATH=$HOME/bin:/usr/local/bin:$PATH
# Path to your oh-my-zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time oh-my-zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
+ ~ head -n20 .zshrc
# Fig pre block. Keep at the top of this file.
[[ -f "$HOME/.fig/shell/zshrc.pre.zsh" ]] && builtin source "$HOME/.fig/shell/zshrc.pre.zsh"
# export PATH=$HOME/bin:/usr/local/bin:$PATH
# Path to your oh-my-zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time oh-my-zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="robbyrussell"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
```

```
anais@MacBook-Pro-de-Anais:~
+ ~ tail .zshrc
# users are encouraged to define aliases within the ZSH_CUSTOM folder.
# For a full list of active aliases, run `alias`.
#
# Example aliases
# alias zshconfig="mate ~/.zshrc"
# alias ohmyzsh="mate ~/.oh-my-zsh"

# Fig post block. Keep at the bottom of this file.
[[ -d "$HOME/.fig/shell/zshrc.post.zsh" ]] && builtin source "$HOME/.fig/shell/zshrc.post.zsh"
test -e "/Users/anais/.item2_shell_integration.zsh" && source "/Users/anais/.item2_shell_integration.zsh" || true
+ ~ tail -n20 .zshrc
# export EDITOR='vim'
# else
# export EDITOR='mvim'
# fi

# Compilation Flags
# export ARCHFLAGS="--arch x86_64"

# Set personal aliases, overriding those provided by oh-my-zsh libs,
# plugins, and themes. Aliases can be placed here, though oh-my-zsh
# users are encouraged to define aliases within the ZSH_CUSTOM folder.
# For a full list of active aliases, run `alias`.
#
# Example aliases
# alias zshconfig="mate ~/.zshrc"
```

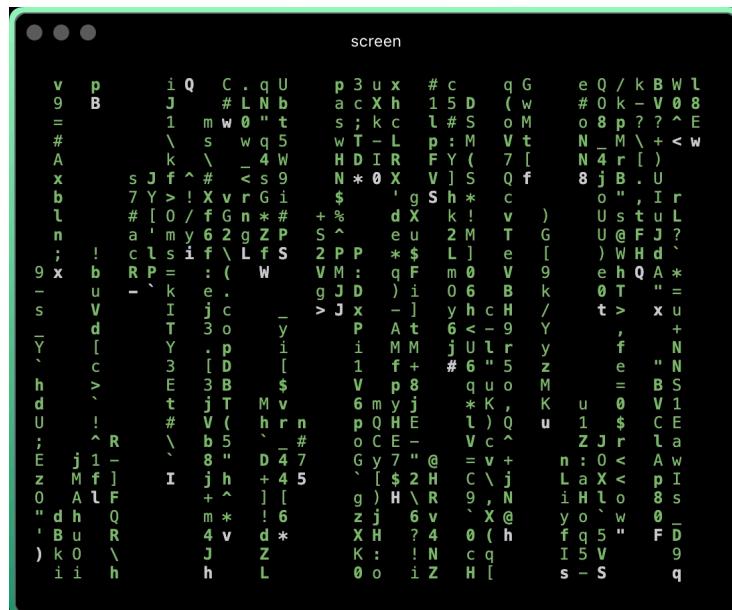
A partir du Job3 les commandes vont commencer à différer entre le Shell zsh et bash. Nous allons donc commencer par faire les Jobs pour MacOS.

MacOS

JOB3

Je vais vous montrer le processus étape par étape pour installer Cmatrix.

1. Installer Homebrew (voir comment [ici](#))
2. Installer cmatrix
`brew install cmatrix`
3. Install & Configure GNU Screen
`brew install homebrew/homebrew-core/screen`
4. Créer un fichier .screenrc
`touch ~/.screenrc`
5. Incrire dans le fichier avec la commande « `nano` »
`blankerprg cmatrix -ab`
`Idle 15 blanker`
6. Ecrire screen dans le terminal et taper sur Entré



```
screen

v p i Q C . q U p 3 u x # c q G e Q / k B W l
9 B J # L N b a c X h 1 5 D ( w # 0 k - V 0 8
= 1 m w 0 " t s ; k c l # S o M o 8 p ? ? ^ E
# \ s w q 5 w T - L p : M V t N _ M \ + < w
A k \ - 4 W H D I R F Y ( 7 [ N 4 r [ )
X s J f ^ # < s 9 N * 0 X V J S Q f 8 j B . U
b 7 Y > ! X v r G i $ ' g S h * c o " , I r
l # [ 0 / f G n * # + % d X k ! v ) U s t u L
n a ' m y 6 2 g Z P S ^ e u 2 M T G U @ F J ?
; ! c l s i f \ L f S 2 P P * $ L ] e [ ) W H d \
9 x b R P = : ( W V M : q F m 0 V 9 e h Q A *
- u - k e . g J D ) i 0 6 B K 0 T " =
s V I j c - > J x - l y h c H / t > x u
d T 3 o y P A t 6 < - 9 Y , f + N
Y [ Y . p i i M M j U l r y e " N
c 3 [ D [ 1 f + # 6 " 5 z e = B S
h > E 3 B $ V p 8 q u o M u 0 V 1
d ` t j T M v 6 m y j * K , K u 1 $ C E
U ! # V ( h r n p Q H E 1 ) Q u 1 Z J r l a
; ^ R \ b 5 ` # o C E - V c ^ n : 0 < A w
E j 1 - 8 " D 4 7 G y 7 " @ = v + L a X < p I
z M f ] I j h + 4 5 ` [ $ 2 H C \ j i H l o 8 s
O A l F + ^ ] [ g ) H \ R 9 , N y o _ w 0
" d h Q m * ! 6 z j 6 v X @ f q 5 " F D
' B u R 4 v d * X H ? 4 0 ( h I 5 V 9
) k 0 \ J Z K : ! N c q s - s q
i i h h L 0 o i Z H [
```

Pour installer les internets (Google) :

Il faut tout d'abord installer wget si ce n'est pas encore fait, voici la commande pour le faire :

brew install wget

Une fois wget installé nous allons l'utiliser pour installer google

wget -r google.com

L'option -r renseigne la récursivité



```
→ ~ wget -r google.com
--2022-09-23 09:43:59-- http://google.com/
Résolution de google.com (google.com)... 216.58.205.206
Connexion à google.com (google.com)|216.58.205.206|:80... connecté.
requête HTTP transmise, en attente de la réponse... 301 Moved Permanently
Emplacement : http://www.google.com/ [suivant]
--2022-09-23 09:43:59-- http://www.google.com/
Résolution de www.google.com (www.google.com)... 172.217.18.36
Connexion à www.google.com (www.google.com)|172.217.18.36|:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : non indiqué [text/html]
Sauvegarde en : « google.com/index.html »

google.com/index.html [=>] 13,62K --.-KB/s ds 0,002s
2022-09-23 09:43:59 (8,31 MB/s) - « google.com/index.html » sauvegardé [1
3945]

Terminé - 2022-09-23 09:43:59 -
Temps total effectif : 0,3s
Téléchargés : 1 fichiers, 14K en 0,002s (8,31 MB/s)
→ ~
```

Nous allons voir maintenant comment redémarrer et éteindre sa machine :

Pour redémarrer

sudo shutdown -r now

Pour éteindre

sudo shutdown -h now

Pour choisir un timer qui l'éteindra dans X mins :

sudo shutdown -h +30

(30 représenté le nombre de minutes souhaité dans cette exemple)

JOB4

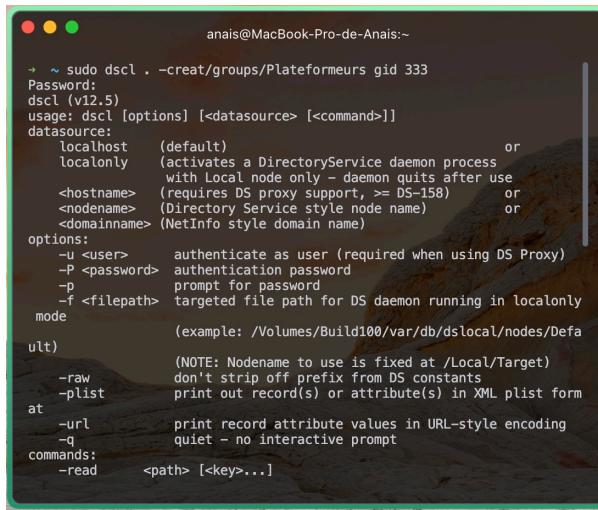
Créer un groupe il faut se mettre en ***sudo*** soit en incluant sudo au début de la ligne soit en tapant ***sudo -s*** puis entrée et ensuite on peut écrire la commande sans l'inclure car nous serrons déjà en sudo

Et du coup la commande pour créer le groupe est ...

sudo dscl . -creat/groups/le nom du groupe

En l'occurrence là le groupe se nomme Plateformeurs il faut aussi lui mettre une ID en ajoutant gid + le numéro de l'ID

sudo dscl . -creat/groups/Plateformeurs gid 333



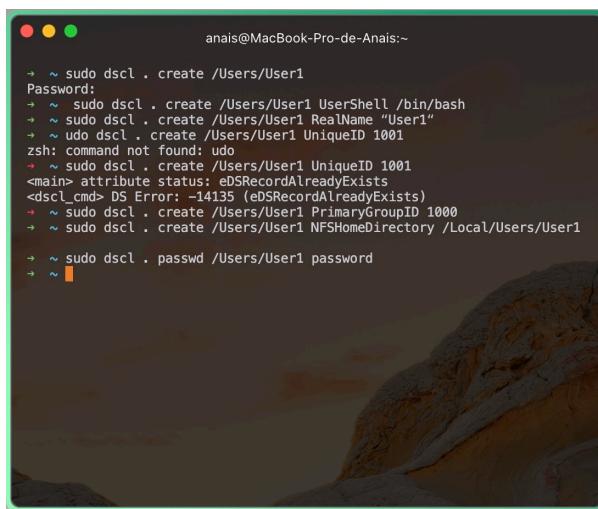
```
anais@MacBook-Pro-de-Anais:~
→ ~ sudo dscl . -creat/groups/Plateformeurs gid 333
Password:
dscl (v12.5)
usage: dscl [options] [<datasource> [<command>]]
datasource:
    localhost      (default)          or
    localonly     (activates a DirectoryService daemon process
                  with Local node only - daemon quits after use)
    <hostname>     (requires DS proxy support, >= DS-158)       or
    <nodename>     (Directory Service style node name)           or
    <domainname>   (NetInfo style domain name)
options:
    -u <user>      authenticate as user (required when using DS Proxy)
    -P <password>  authentication password
    -p              prompt for password
    -f <filepath>   targeted file path for DS daemon running in localonly
mode
    (example: /Volumes/Build100/var/db/dslocal/nodes/Defau
ult)
    -raw            (NOTE: Nodename to use is fixed at /Local/Target)
    -plist          don't strip off prefix from DS constants
    -at             print out record(s) or attribute(s) in XML plist form
    -url            print record attribute values in URL-style encoding
    -q              quiet - no interactive prompt
commands:
    -read           <path> [<key>...]
```

Maintenant on va créer un utilisateur qui s'appellera User1 :

Il faudra à lui aussi lui créer une ID toujours en mode sudo.
Voici les différentes commandes qu'il faudra faire :

1. *sudo dscl . create /Users/User1*
2. *sudo dscl . create /Users/User1 UserShell /bin/bash*
3. *sudo dscl . create /Users/User1 RealName "User1"*
4. *sudo dscl . create /Users/User1 UniqueID 1001*
5. *sudo dscl . create /Users/User1 PrimaryGroupID 1000*
6. *sudo dscl . create /Users/User1 NFSHomeDirectory /Local/Users/User1*
7. *sudo dscl . passwd /Users/User1 password*

Et nous allons aussi créer un copain qui lui s'appellera User2



```
anais@MacBook-Pro-de-Anais:~
→ ~ sudo dscl . create /Users/User1
Password:
→ ~ sudo dscl . create /Users/User1 UserShell /bin/bash
→ ~ sudo dscl . create /Users/User1 RealName "User1"
→ ~ sudo dscl . create /Users/User1 UniqueID 1001
zsh: command not found: sudo
→ ~ sudo dscl . create /Users/User1 UniqueID 1001
<main> attribute status: eDSRecordAlreadyExists
<dscl_cmd> DS Error: -14135 (eDSRecordAlreadyExists)
→ ~ sudo dscl . create /Users/User1 PrimaryGroupID 1000
→ ~ sudo dscl . create /Users/User1 NFSHomeDirectory /Local/Users/User1
→ ~ sudo dscl . passwd /Users/User1 password
→ ~
```

Il est demandé dans la consigne d'ajouter le User2 dans le groupe Plateformeurs, il faudra toujours rester en sudo et écrire la commande suivante :

```
sudo dseditgroup -o edit -a User2 -t user Plateformeurs
```

dseditgroup permet la manipulation d'un enregistrement de groupe nommé sur le nœud local par défaut

-o opération effectuer (lire, créer, supprimer, modifier, vérifier le membre)

-a recordname nom de l'enregistrement à ajouter

-t recordtype type de l'enregistrement à ajouter ou à supprimer

Pour vérifier qu'il y soit vraiment on utilisera :

```
dscacheutil -q group -a name Plateformeurs
```

-q est pour la catégorie et le -a pour le nom du groupe

Pour supprimer un utilisateur d'un groupe se sera :

```
sudo dscl . -delete /Groups/Plateformeurs
```

Pour copier un fichier dans un autre il suffit de faire :

```
cp fichier1 fichier2
```

```
cp users.txt droits.txt
```

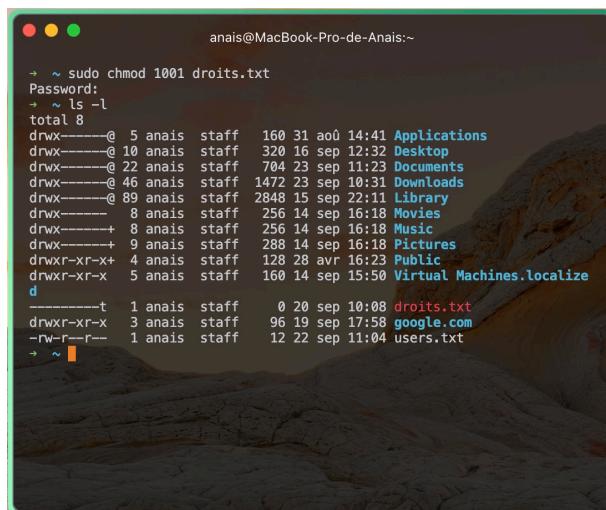
```
cp users.txt groups.txt
```

Pour changer le propriétaire d'un fichier, nous allons utiliser la commande chmod (

Sudo chown + l'ID de l'user + fichier

Sudo chown 1001 droits.txt

Pour vérifier que le propriétaire a été changé il faut faire : ls -l



The screenshot shows a terminal window on a Mac OS X desktop. The command entered was `sudo chmod 1001 droits.txt`. The terminal then lists the contents of the current directory with the command `ls -l`. The output shows the following file listing:

```
anais@MacBook-Pro-de-Anais:~  
→ ~ sudo chmod 1001 droits.txt  
Password:  
→ ~ ls -l  
total 8  
drwx-----@ 5 anais staff 160 31 aoû 14:41 Applications  
drwx-----@ 10 anais staff 320 18 sep 12:32 Desktop  
drwx-----@ 22 anais staff 704 23 sep 11:23 Documents  
drwx-----@ 46 anais staff 1472 23 sep 10:31 Downloads  
drwx-----@ 89 anais staff 2848 15 sep 22:11 Library  
drwx-----@ 8 anais staff 256 14 sep 16:18 Movies  
drwx-----@ 8 anais staff 256 14 sep 16:18 Music  
drwx-----@ 9 anais staff 288 14 sep 16:18 Pictures  
drwxr-xr-x+ 4 anais staff 128 28 avr 16:23 Public  
drwxr-xr-x  5 anais staff 160 14 sep 15:50 Virtual Machines.localized  
d-----t 1 anais staff 0 20 sep 10:08 droits.txt  
drwxr-xr-x  3 anais staff 96 19 sep 17:58 google.com  
-rw-r--r--  1 anais staff 12 22 sep 11:04 users.txt  
→ ~
```

>

Maintenant on va changer les droits du fichiers droits.txt pour que l'User2 y ai accès qu'en mode lecture

La commande `chmod` permet de modifier droits et permissions.

`sudo chmod 744 droits.txt`

Dans cette commande le premier chiffre correspond aux droits du propriétaire
7 correspond à rwx qui donne le droit

Le deuxième chiffre correspond aux droits du groupe
4 correspond à r - - qui donne le droit à la lecture seulement

Le troisième chiffre correspond aux droits des autres
4 correspond à r - - qui donne le droit à la lecture seulement

```
anais@MacBook-Pro-de-Anais:~
→ ~ sudo chmod 744 droits.txt
Password:
→ ~ ls -l
total 8
drwxr-xr-x  2 anais  staff   160 31 aoû 14:41 Applications
drwxr-xr-x  2 anais  staff   320 16 sep 12:32 Desktop
drwxr-xr-x  2 anais  staff   704 23 sep 11:35 Documents
drwxr-xr-x  2 anais  staff  1504 23 sep 11:29 Downloads
drwxr-xr-x  2 anais  staff  2848 15 sep 22:11 Library
drwxr-xr-x  2 anais  staff   256 14 sep 16:18 Movies
drwxr-xr-x  2 anais  staff   256 14 sep 16:18 Music
drwxr-xr-x  2 anais  staff   288 14 sep 16:18 Pictures
drwxr-xr-x  2 anais  staff   128 28 avr 16:23 Public
drwxr-xr-x  2 anais  staff  160 14 sep 15:50 Virtual Machines.localized
d
drwxr-xr-x  1 anais  staff     0 20 sep 10:08 droits.txt
drwxr-xr-x  3 anais  staff    96 19 sep 17:58 google.com
drwxr-xr-x  1 anais  staff   12 22 sep 11:04 users.txt
→ ~
```

II - Correspondances des droits en binaire/octale et leurs significations

- **u** signifie : user (propriétaire) ;
 - **g** signifie : group (groupe) ;
 - **o** signifie : other (autres) ;
- ... et que :
- **+** signifie : « Ajouter le droit » ;
 - **-** signifie : « Supprimer le droit » ;
 - **=** signifie : « Affecter le droit ».

Position Binaire	Valeur octale	Droits	Signification
000	0	---	Aucun droit
001	1	--x	Exécutable
010	2	-w-	Ecriture
011	3	-wx	Ecrire et exécuter
100	4	r--	Lire
101	5	r-x	Lire et exécuter
110	6	r-w-	Lire et écrire
111	7	rwx	Lire écrire et exécuter

Pour vérifier que les droits on était modifier il faut faire `ls -al` et regarder la première colonne

Voir les sites qui suivent pour plus de détails [Site1](#) [Site2](#) [Site3](#) [Site4](#)

On fera de même pour le fichier groupes.txt

`sudo chmod 744 groupes.txt`

Et pour le groupe Plateformeurs on changera le deuxième chiffre qui correspond au droits du groupe

`sudo chmod 764 Platefomeurs`

JOB5

Nous allons faire un alias pour lancer la commande ls -la en tapant "la"
Un alias c'est ce qui va permet de créer un raccourci pour une commande.

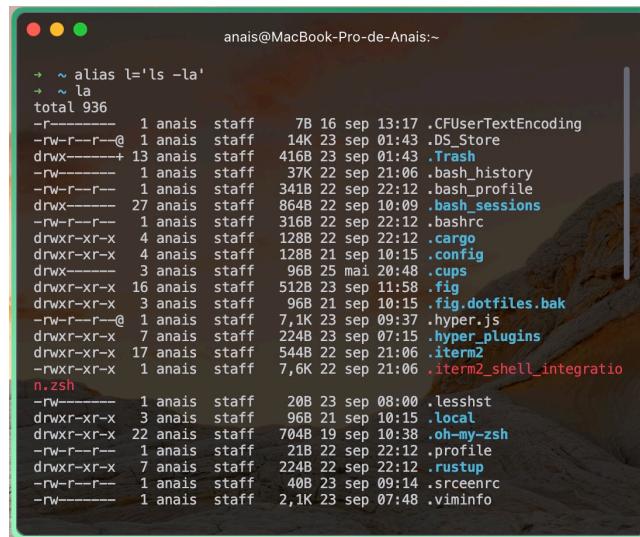
`alias la='ls -la'`

Il suffit de faire la commande alias avec le nom qu'on souhaite donner au raccourci -la + la commande qu'on souhaite réduire

Plus de détail [ici](#)

`alias update='brew update'`

`alias upgrade='brew upgrade'`

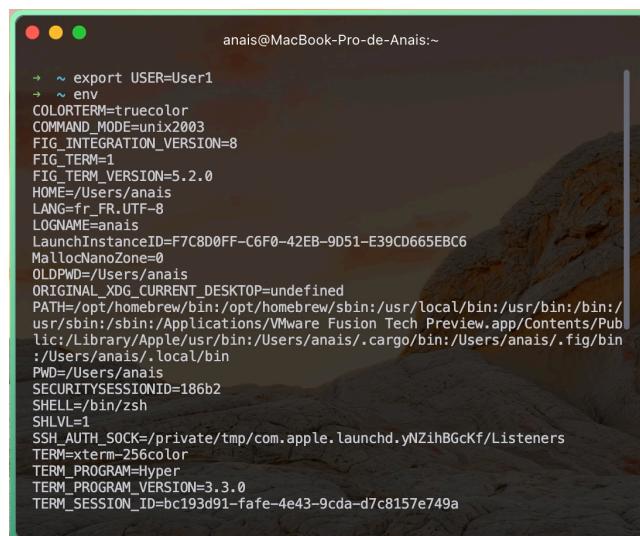


```
anais@MacBook-Pro-de-Anais:~  
$ ~ alias l='ls -la'  
$ ~ la  
total 936  
-r----- 1 anais staff 7B 16 sep 13:17 .CFUserTextEncoding  
-rw-r--r--@ 1 anais staff 14K 23 sep 01:43 .DS_Store  
drwx-----+ 13 anais staff 416B 23 sep 01:43 .Trash  
-rw------- 1 anais staff 37K 22 sep 21:06 .bash_history  
-rw-r--r-- 1 anais staff 341B 22 sep 22:12 .bash_profile  
drwx----- 27 anais staff 864B 22 sep 10:09 .bash_sessions  
-rw-r--r-- 1 anais staff 316B 22 sep 22:12 .bashrc  
drwxr-xr-x 4 anais staff 128B 22 sep 22:12 .cargo  
drwxr-xr-x 4 anais staff 128B 21 sep 10:15 .config  
drwx----- 3 anais staff 96B 25 mai 20:48 .cups  
drwxr-xr-x 16 anais staff 512B 23 sep 11:58 .fig  
drwxr-xr-x 3 anais staff 96B 21 sep 10:15 .fig.dotfiles.bak  
-rw-r--r--@ 1 anais staff 7,1K 23 sep 09:37 .hyper.js  
drwxr-xr-x 7 anais staff 224B 23 sep 07:15 .hyper_plugins  
drwxr-xr-x 17 anais staff 544B 22 sep 21:06 .item1  
-rwxr-xr-x 1 anais staff 7,6K 22 sep 21:06 .item2_shell_integration  
n.zsh  
-rw------- 1 anais staff 20B 23 sep 08:00 .lessht  
drwxr-xr-x 3 anais staff 96B 21 sep 10:15 .local  
drwxr-xr-x 22 anais staff 704B 19 sep 10:38 .oh-my-zsh  
-rw-r--r-- 1 anais staff 21B 22 sep 22:12 .profile  
drwxr-xr-x 7 anais staff 224B 22 sep 22:12 .rustup  
-rw-r--r-- 1 anais staff 40B 23 sep 09:14 .srceenrc  
-rw----- 1 anais staff 2,1K 23 sep 07:48 .viminfo
```

Pour créer une variable d'environnement temporaire on utilise :

`export USER=User1`

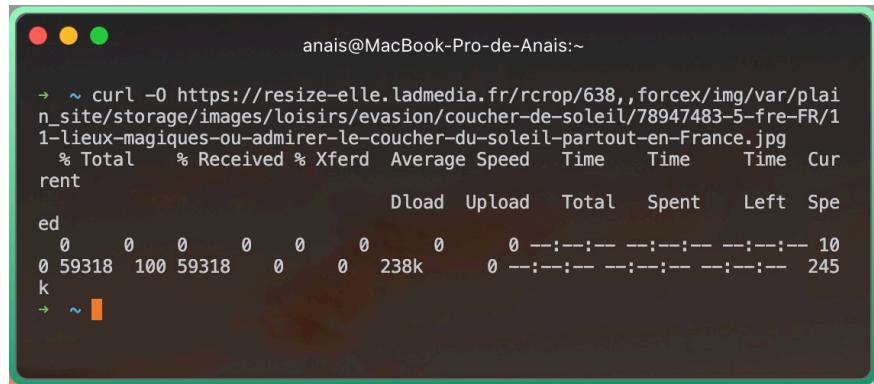
Pour voir la variable d'environnement il faut écrire : env



```
anais@MacBook-Pro-de-Anais:~  
$ ~ export USER=User1  
$ ~ env  
COLORTERM=truecolor  
COMMAND_MODE=unix2003  
FIG_INTEGRATION_VERSION=8  
FIG_TERM=1  
FIG_TERM_VERSION=5.2.0  
HOME=/Users/anais  
LANG=fr_FR.UTF-8  
LOGNAME=anais  
LaunchInstanceId=F7C8D0FF-C6F0-42EB-9D51-E39CD665EBC6  
MallocNanoZone=0  
OLDPWD=/Users/anais  
ORIGINAL_XDG_CURRENT_DESKTOP=undefined  
PATH=/opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Applications/VMware Fusion Tech Preview.app/Contents/Public:/Library/Application/Usr/bin:/Users/anais/.cargo/bin:/Users/anais/.fig/bin:/Users/anais/.local/bin  
PWD=/Users/anais  
SECURITYSESSIONID=186b2  
SHELL=/bin/zsh  
SHLVL=1  
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.yNZihBGcKf/Listeners  
TERM=xterm-256color  
TERM_PROGRAM=Hyper  
TERM_PROGRAM_VERSION=3.3.0  
TERM_SESSION_ID=bc193d91-fafe-4e43-9cda-d7c8157e749a
```

> JOB6

Pour télécharger une image il faut utiliser la commande
curl -O + le lien de la photo



```
anaïs@MacBook-Pro-de-Anais:~  
→ ~ curl -O https://resize-elle.ladmedia.fr/rcrop/638,,forceX/img/var/plai  
n_site/storage/images/loisirs/evasion/coucher-de-soleil/78947483-5-fre-FR/1  
1-lieux-magiques-ou-admire-le-coucher-du-soleil-partout-en-France.jpg  
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current  
                                         Dload  Upload   Total   Spent    Left  Speed  
ed  
 0      0      0      0      0      0      0      0 --:--:-- --:--:-- --:--:-- 10  
0 59318 100 59318    0      0  238k      0 --:--:-- --:--:-- --:--:-- 245  
k  
→ ~ |
```

-O permet de spécifier l'*output*, ce qui veut dire qu'il va télécharger le fichier.

Si le fichier est compressé on ajoutera la commande tar
`-xvf nomdufichier.tgz`

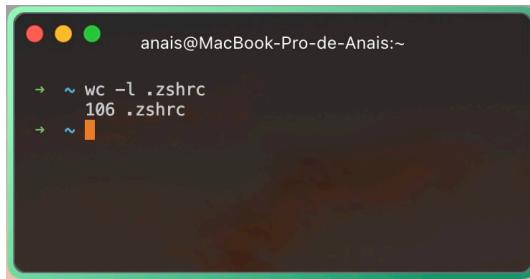
Le paramètre x permet d'extraire l'archive,
Le z de la décompresser,
Le f indique la donnée à décompresser

JOB7

Pour créer un fichier tout en écrivant du texte à l'intérieur on utilise `echo`
`echo "Je suis votre fichier texte" > une_commande.txt`

Pour connaitre le nombre de ligne présente dans le fichier sources (qui est `.zshrc`) on fait :

`wc -l .zshrc`
(J'en ai 106)



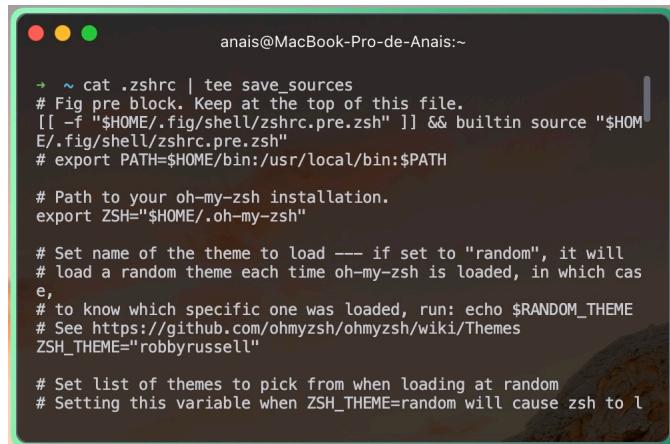
```
anaïs@MacBook-Pro-de-Anais:~  
→ ~ wc -l .zshrc  
106 .zshrc  
→ ~
```

Si nous voulons inscrire le résultat directement dans un fichier, nous faisons la même manip mais en rajoutant un chevron qui dirigera le résultat direct dans le fichier que nous indiquerons.

`wc -l .zshrc > nb_lignes.txt`

Pour afficher le contenu d'un fichier on utilise la commande cat mais là nous allons directement la copier dans un autre fichier.

`cat .zshrc | tee save_sources`



```
anaïs@MacBook-Pro-de-Anais:~  
→ ~ cat .zshrc | tee save_sources  
# Fig pre block. Keep at the top of this file.  
[[ -f "$HOME/.fig/shell/zshrc.pre.zsh" ]] && builtin source "$HOME/.fig/shell/zshrc.pre.zsh"  
# export PATH=$HOME/bin:/usr/local/bin:$PATH  
  
# Path to your oh-my-zsh installation.  
export ZSH="$HOME/.oh-my-zsh"  
  
# Set name of the theme to load --- if set to "random", it will  
# load a random theme each time oh-my-zsh is loaded, in which cas  
e,  
# to know which specific one was loaded, run: echo $RANDOM_THEME  
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes  
ZSH_THEME="robbyrussell"  
  
# Set list of themes to pick from when loading at random  
# Setting this variable when ZSH_THEME=random will cause zsh to l
```

Pour trouver un fichier qui commence par une lettre en particulier on utilisera :

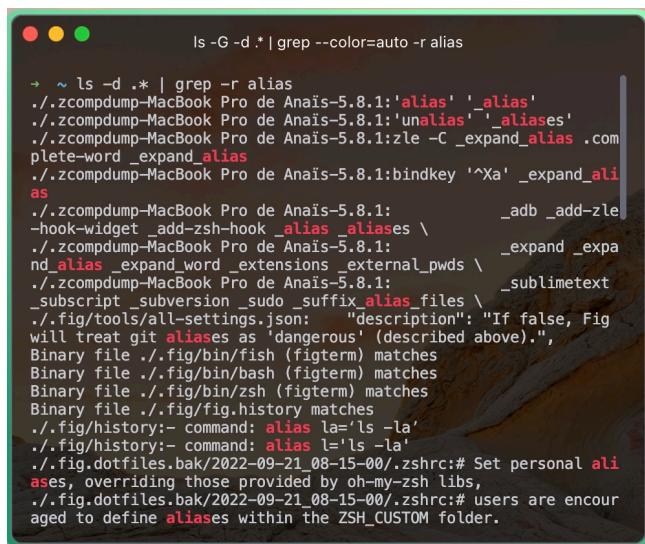
`ls -d a*`

Ici se sera :

`ls -d .*`

Et pour lister tous les fichiers commençant pas un « . » avec le mot « alias » à l'intérieur, on ferra :

`ls -d .* | grep -r alias`



```
ls -G -d .* | grep --color=auto -r alias
./.zcompdump-MacBook Pro de Anaïs-5.8.1:'alias' '_alias'
./.zcompdump-MacBook Pro de Anaïs-5.8.1:'unalias' '_aliases'
./.zcompdump-MacBook Pro de Anaïs-5.8.1:zle -C _expand_alias .com
plete-word _expand_alias
./.zcompdump-MacBook Pro de Anaïs-5.8.1:bindkey '^Xa' _expand_ali
as
./.zcompdump-MacBook Pro de Anaïs-5.8.1: _adb _add-zle
-hook-widget _add-zsh-hook _alias _aliases \
./.zcompdump-MacBook Pro de Anaïs-5.8.1: _expand _expa
nd _alias _expand_word _extensions _external_pwds \
./.zcompdump-MacBook Pro de Anaïs-5.8.1: _sublimetext
_subscript _subversion _sudo _suffix _alias_files \
./.fig/tools/all-settings.json: "description": "If false, Fig
will treat git aliases as 'dangerous' (described above).",
Binary file ./fig/bin/fish (figterm) matches
Binary file ./fig/bin/bash (figterm) matches
Binary file ./fig/bin/zsh (figterm) matches
Binary file ./fig/fig.history matches
./.fig/history:- command: alias la='ls -la'
./.fig/history:- command: alias l='ls -la'
./.fig.dotfiles.bak/2022-09-21_08-15-00/.zshrc:# Set personal ali
ases, overriding those provided by oh-my-zsh libs,
./.fig.dotfiles.bak/2022-09-21_08-15-00/.zshrc:# users are encour
aged to define aliases within the ZSH_CUSTOM folder.
```

Pour aller plus loin...

Pour installer tree on devons utiliser :

`brew install tree`

Quand nous souhaitons lancer un fichier en arrière plan on devons mettre « & » à la fin de la commande

Ici pour nous allons lancer le fichier en arrière plan et le copier le contenu directement dans un autre fichier :

`tree > tree.save &`

Si nous lister le nombre d'éléments du dossier courant on utilise la commande

`ls | wc -l`

On peut vérifier le fichier en tapant :

`cat tree.save`

Pour faire une mise à jour on utilise update et upgrade mais pour le faire en une seule ligne on fait :

`update && upgrade`

&& : signifie exécuter l'instruction qui suit *uniquement si* l'instruction précédente s'est exécutée avec succès

& : signifie exécuter l'instruction précédente en arrière-plan

|| : signifie exécuter l'instruction qui suit *uniquement si* l'instruction précédente a échoué

| : signifie exécuter l'instruction précédente et connecter sa stdout à stdin de l'instruction qui suit.

; : signifie exécuter l'instruction précédente et, une fois terminée, passer à l'instruction suivante.

Linux

JOB3

Pour installer cmatrix on utilisera le mode sudo avec apt-get

sudo apt-get install cmatrix

Puis il faudra écrire dans le terminal cmatrix et taper sur Entrée



The terminal window shows a complex pattern of ASCII characters forming a matrix-style display. The title bar reads "debian@debian: ~". The menu bar includes "File", "Edit", "Tabs", and "Help". The command line is empty, indicated by a tilde (~).

Pour redémarrer ou éteindre sa machine depuis le terminal on utilisera :

Redémarrer : **systemctl halt**

Eteindre : **systemctl poweroff**

Choisir a quelle heure on souhaite l'éteindre : **shutdown - -poweroff 23:00**

JOB4

Lorsque que nous voulons créer un groupe on fera :

`sudo groupadd Plateformeurs`

Pour retrouver un groupe :

`getent group`

Il est aussi possible de créer plusieurs groupes en même temps en utilisant :

`sudo groupadd namegroup1, namegroup2, namegroup3`

Pour supprimer un groupe :

`sudo groupdel namegroup`

Créer un utilisateur appelé “User1”

1. Mettre son Shell en root

`sudo -s`

Pour quitter le mode root : `exit`

2. Créer un utilisateur

`useradd nameuser`

Retrouver son nouvel utilisateur :

Pour trouver les utilisateurs que nous venons de créer il faut faire la même commande que

pour retrouver le groupe créé : `getent groups`

Créer un utilisateur et de lui attribuer des groupes en même temps :

Etre en root (`sudo -s`)

`useradd -G groupe1, groupe2, groupe3 nameuser`

Supprimer un utilisateur :

Etre en root (`sudo -s`)

`userdel nameuser`

Créer un utilisateur appelé “User2”

La même méthode que pour le user1

Ajouter “User2” au groupe Plateformeurs

Pour ajouter un utilisateur dans un groupe :

`sudo usermod -a -G Plateformeurs User2`

Pour vérifier que la manipulation a fonctionné :

`getent group`

Ca affichera tous les groupes et ce qu'ils contiennent

Copier votre “users.txt” dans un fichier “droits.txt”

Je n'ai pas compris cette consigne car il me semble qu'on ne peut pas copier un fichier dans un autre. J'en déduit donc qu'il faut en fait copier le contenu de users.txt dans droits.txt

cat users.txt >> droit.txt

Copier votre “users.txt” dans un fichier “groupes.txt”

La même manip que juste au-dessus.

cat users.txt >> groupes.txt

Changer le propriétaire du fichier “droits.txt” pour mettre “User1”

Pour changer le propriétaire d'un fichier il faut utiliser la commande : *chown*

*Sudo chown utilisateur fichier
Sudo chown User1 droits.txt*

Pour vérifier que le propriétaire à été changer il faut faire : *ls -l*

Changer les droits du fichier “droits.txt” pour que “User2” ai accès seulement en lecture

Ici se sera la même manip que pour zsh sur MacOS (voir [ici](#))

Changer les droits du fichier “groupes.txt” pour que les utilisateurs puissent accéder au fichier en lecture uniquement

sudo chmod 744 groupes.txt

Changer les droits du fichier pour que le groupe “Platefomeurs” puissent y accéder en lecture/écriture

sudo chmod 764 Platefomeurs

JOB5

Ajouter un alias qui permettra de lancer la commande “ls -la” en tapant “la”

alias la='ls -la'

Il suffit de faire la commande alias avec le nom qu'on souhaite donner au raccourci -la commande qu'on souhaite réduire

Plus de détail [ici](#)

Ajouter un alias qui permettra de lancer la commande “apt-get update” en tapant “update”

alias update='apt-get update'

Ajouter un alias qui permettra de lancer la commande “apt-get upgrade” en tapant “upgrade”

alias upgrade='apt-get upgrade'

Ajouter une variable d'environnement qui se nommera “USER” et qui sera égale à votre nom d'utilisateur

La commande pour une variable d'environnement est : *env*

Pour le nommé il faut faire : *env NOM*

Pour qu'il soit égal à notre nom d'utilisateur il faut faire : *env
NOM='nomdesonuser'*

env USER='debian'

(debian est mon nom d'utilisateur)

Plus de détail [ici](#)

Mettre à jour les modifications de votre bashrc dans votre shell actuel

Pour le mettre à jour il faut l'ouvrir avec nano faire des modifications et l'enregistrer

nano ~/.bashrc

Afficher les variables d'environnement

Pour afficher les variable d'environnement : *env* ou *printenv*

Sans aucun argument, la commande *set* répertorie toutes les variables shell, y compris leurs valeurs.

Lien :

[Comment lire et configurer les variables d'environnement et de shell sous Linux](#)

Ajouter à votre Path le chemin "/home/'votre utilisateur'/Bureau"

Pour ajouter un chemin a son PATH il faut utiliser :

export PATH=\$PATH:/chemin/vers/lerépertoire

Ce qui donne ici :

export PATH=\$PATH:/home/debian/Bureau

Plus de détail [ici](#)

JOB6

Ici aussi les commande seront les même que pour le Shell zsh de MacOS

Pour y jeter un œil à nouveau c'est par > [ici](#)

JOB7

Créer un fichier “une_commande.txt” avec le texte suivant “Je suis votre fichier texte”

echo “Je suis votre fichier texte.” > une_commande.txt

echo va permettre d'insérer le direct directement et le chevron lui servira à diriger ce texte dans le fichier.

Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé “nb_lignes.txt”

Pour commencer, le fichier source apt se nomme sources.list ou sources.list.d
Son chemin c'est /etc/apt/sources.list.d

Il y a 2 manière de trouver le nombre de ligne qui s'y trouve :

wc -l /etc/apt/sources.list

qui va afficher ses droits

ls -lh /etc/apt/sources.list

qui va en revanche seulement indiquer le nombre de ligne

(J'en ai 21 ici)

Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé “save_sources”

La commande pour afficher le contenu d'un fichier c'est **cat**

Donc pour afficher le contenu du fichier source apt il faut :
cat /etc/apt/sources.list

Pour copier le contenu et le mettre dans un autre fichier il suffit de faire :
cat /etc/apt/sources.list >> save_sources.txt

Faites une recherche des fichiers commençants par “.” tout en cherchant le mot alias qui sera utilisé depuis un fichier

ls -d .* | grep -r alias

Cette commande sera la même que celle pour le Shell zsh de MacOS

Tout savoir sur la commande grep : [ici](#)

Pour aller plus loin...

Installer la commande tree

```
apt-get install tree  
apt update  
apt upgrade
```

Lancer la commande tree en arrière-plan qui aura pour but d'afficher toute l'arborescence en de votre / en enregistrant le résultat dans un fichier “tree.save”

Afin de lister les fichiers et sous dossier d'un répertoire on utilise :

tree -a [Nom du répertoire/Chemin]

Le « / » représente la racine, j'ai donc fait :

tree -a /

Pour enregistrer le résultat dans le fichier tree.save il suffit d'utiliser le chevrons >> + le fichier souhaité

tree -a / > tree.save

Le **>>** écrit à la fin d'un fichier

Le **>** efface tout et réécrit

**lister les éléments présents dans le dossier courant est utilisé directement
le résultat de votre première commande pour compter le nombre
d'éléments trouvés**

Pour afficher le nombre d'élément il faut faire « **-n** »

Ca donne donc :

ls -la -n

Mais on peut aussi faire :

ls -lan

(Il indique « total 13216 »)

**Lancer une commande pour update vos paquets, si l'update réussi alors,
vous devrez lancer un upgrade de vos paquets. Si l'update échoue, votre
upgrade ne se lancera pas**

apt update && apt upgrade