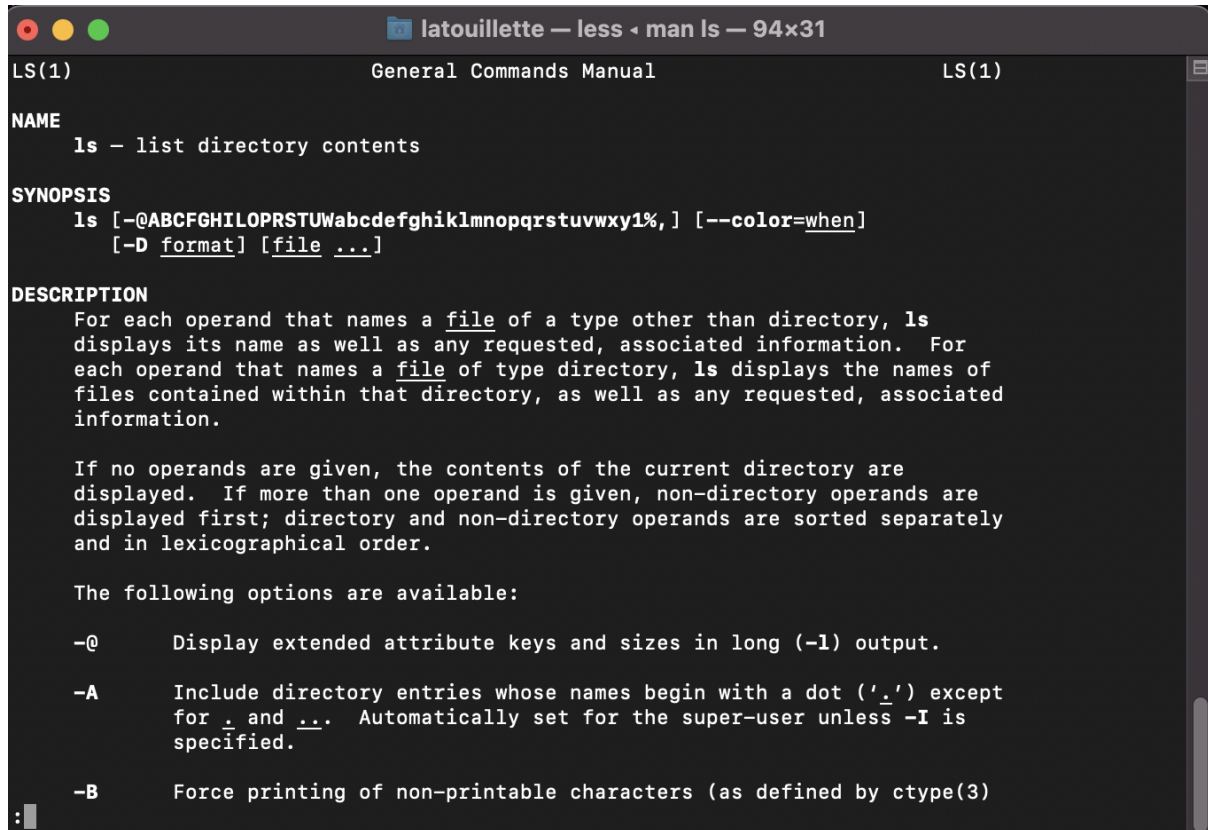


## SHELL

JOB 1 :

### Afficher le manuel de la commande ls :

il faut utiliser `man ls` qui donne en sortie le manuel de la commande `ls`



```
LS(1)                                General Commands Manual                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [-@ABCFGHILOPRSTUWabcdefghiklmnopqrstuvwx1%,] [--color=when]
    [-D format] [file ...]

DESCRIPTION
  For each operand that names a file of a type other than directory, ls
  displays its name as well as any requested, associated information.  For
  each operand that names a file of type directory, ls displays the names of
  files contained within that directory, as well as any requested, associated
  information.

  If no operands are given, the contents of the current directory are
  displayed.  If more than one operand is given, non-directory operands are
  displayed first; directory and non-directory operands are sorted separately
  and in lexicographical order.

  The following options are available:

  -@      Display extended attribute keys and sizes in long (-l) output.

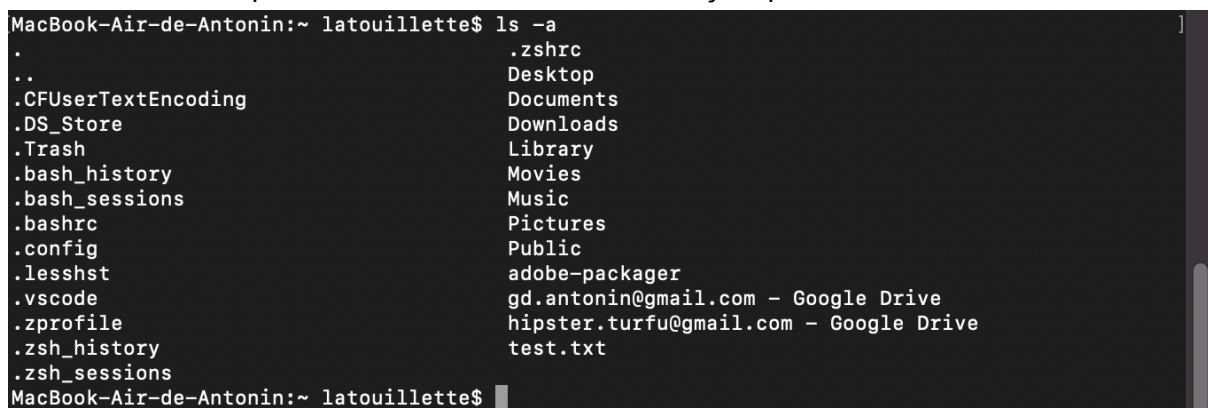
  -A      Include directory entries whose names begin with a dot ('.') except
  for . and ..  Automatically set for the super-user unless -I is
  specified.

  -B      Force printing of non-printable characters (as defined by ctype(3))

:|
```

### Afficher les fichiers cachés du home de votre utilisateur :

il faut utiliser `ls -a` qui affiche aussi les fichier commençant par `.` et `..`



```
MacBook-Air-de-Antonin:~ latouillette$ ls -a
.      .zshrc
..     Desktop
.CFUserTextEncoding  Documents
.DS_Store             Downloads
.Trash               Library
.bash_history         Movies
.bash_sessions       Music
.bashrc              Pictures
.config              Public
.lessshst            adobe-packager
.vscode              gd.antonin@gmail.com - Google Drive
.zprofile            hipster.turfu@gmail.com - Google Drive
.zsh_history         test.txt
.zsh_sessions

MacBook-Air-de-Antonin:~ latouillette$
```

## Afficher les fichiers cachés plus les informations sur les droits sous forme de liste :

il faut utiliser **ls -la** qui affiche, de gauche à droite, les droits (lecture, écriture, exécution) par triplets user groupes et others, le nombre de liens qui mènent au fichier, les propriétaires des fichiers, le groupe du propriétaire, la taille du fichier, sa date de création et son nom.

```
MacBook-Air-de-Antonin:~ latouillette$ ls -la
total 80
drwxr-x---+ 27 latouillette  staff    864  3 oct 11:15 .
drwxr-xr-x   5 root          admin    160 16 sep 16:52 ..
-r-----   1 latouillette  staff     7 16 sep 16:53 .CFUserTextEncoding
-rw-r--r--@  1 latouillette  staff 10244  1 oct 15:01 .DS_Store
drwx-----+ 46 latouillette  staff  1472  1 oct 18:15 .Trash
-rw-----   1 latouillette  staff   539  3 oct 11:09 .bash_history
drwx----- 22 latouillette  staff   704  3 oct 11:09 .bash_sessions
-rw-r--r--@  1 latouillette  staff  3783  3 oct 10:58 .bashrc
drwxr-xr-x   3 latouillette  staff    96 16 sep 17:14 .config
-rw-----   1 latouillette  staff    20  3 oct 11:15 .lessht
drwxr-xr-x   4 latouillette  staff   128 18 sep 23:25 .vscode
-rw-r--r--   1 latouillette  staff    83  3 oct 11:13 .zprofile
-rw-----   1 latouillette  staff  2819  3 oct 10:29 .zsh_history
drwx----- 33 latouillette  staff  1056  3 oct 10:29 .zsh_sessions
-rw-r--r--   1 latouillette  staff     0  3 oct 10:13 .zshrc
drwx-----+  4 latouillette  staff   128  1 oct 15:00 Desktop
drwx-----+ 10 latouillette  staff   320  1 oct 14:57 Documents
drwx-----+ 24 latouillette  staff   768  1 oct 15:31 Downloads
drwx-----@ 82 latouillette  staff  2624  1 oct 14:57 Library
drwx-----   4 latouillette  staff   128 16 sep 17:26 Movies
drwx-----+  4 latouillette  staff   128 16 sep 20:09 Music
drwx-----+  4 latouillette  staff   128 16 sep 16:53 Pictures
drwxr-xr-x+  4 latouillette  staff   128 16 sep 16:52 Public
drwxr-xr-x   8 latouillette  staff   256  1 oct 14:38 adobe-packager
lrwx-----   1 latouillette  staff    73  3 oct 10:46 gd.antonin@gmail.com - Google Drive -> /
Users/latouillette/Library/CloudStorage/GoogleDrive-gd.antonin@gmail.com
lrwx-----   1 latouillette  staff    76  3 oct 10:46 hipster.turfu@gmail.com - Google Drive -
> /Users/latouillette/Library/CloudStorage/GoogleDrive-hipster.turfu@gmail.com
-rw-r--r--@  1 latouillette  staff    89  3 oct 10:22 test.txt
MacBook-Air-de-Antonin:~ latouillette$
```

## Comment ajouter des options à une commande ?

Il suffit de regarder les options disponibles dans le manuel de la commande et de les lister à la suite de l'appel de la commande (souvent précédé de -).

## Quelles sont les deux syntaxes principales d'écriture des options pour une commande ?

On peut mettre un tiret pour chaque option (**ls -l -a**) ou on peut en mettre qu'un seul et mettre toutes les options à la suite (**ls -la**).

JOB 2 :

### Lisez un fichier en utilisant une commande qui permet seulement de lire

il faut utiliser **cat** qui affiche dans le terminal tout le contenu d'un fichier

```
MacBook-Air-de-Antonin:~ latouillette$ 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.
shopt -s checkwinsize

# If set, the pattern "*" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
```

### afficher les 10 premières lignes du fichier ".bashrc"

il faut utiliser **head** qui par défaut affiche dans le terminal les 10 première lignes d'un fichier.

```
MacBook-Air-de-Antonin:~ latouillette$ head .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

MacBook-Air-de-Antonin:~ latouillette$
```

### afficher les 10 dernières lignes du fichier ".bashrc"

il faut utiliser **tail** qui par défaut affiche dans le terminal les 10 dernières lignes d'un fichier.

```
MacBook-Air-de-Antonin:~ latouillette$ tail .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
fiMacBook-Air-de-Antonin:~ latouillette$
```

### afficher les 20 premières lignes du fichier “.bashrc”

il faut utiliser **head -n 20 .bashrc** avec l'option -n x qui permet de paramétrer le nombre de lignes affichées.

```
fiMacBook-Air-de-Antonin:~ latouillette$ 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
MacBook-Air-de-Antonin:~ latouillette$
```

### afficher les 20 dernières lignes du fichier “.bashrc”

il faut utiliser **tail -n 20 .bashrc** avec l'option -n x qui permet de paramétrer le nombre de lignes affichées.

```
MacBook-Air-de-Antonin:~ latouillette$ 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
fiMacBook-Air-de-Antonin:~ latouillette$
```

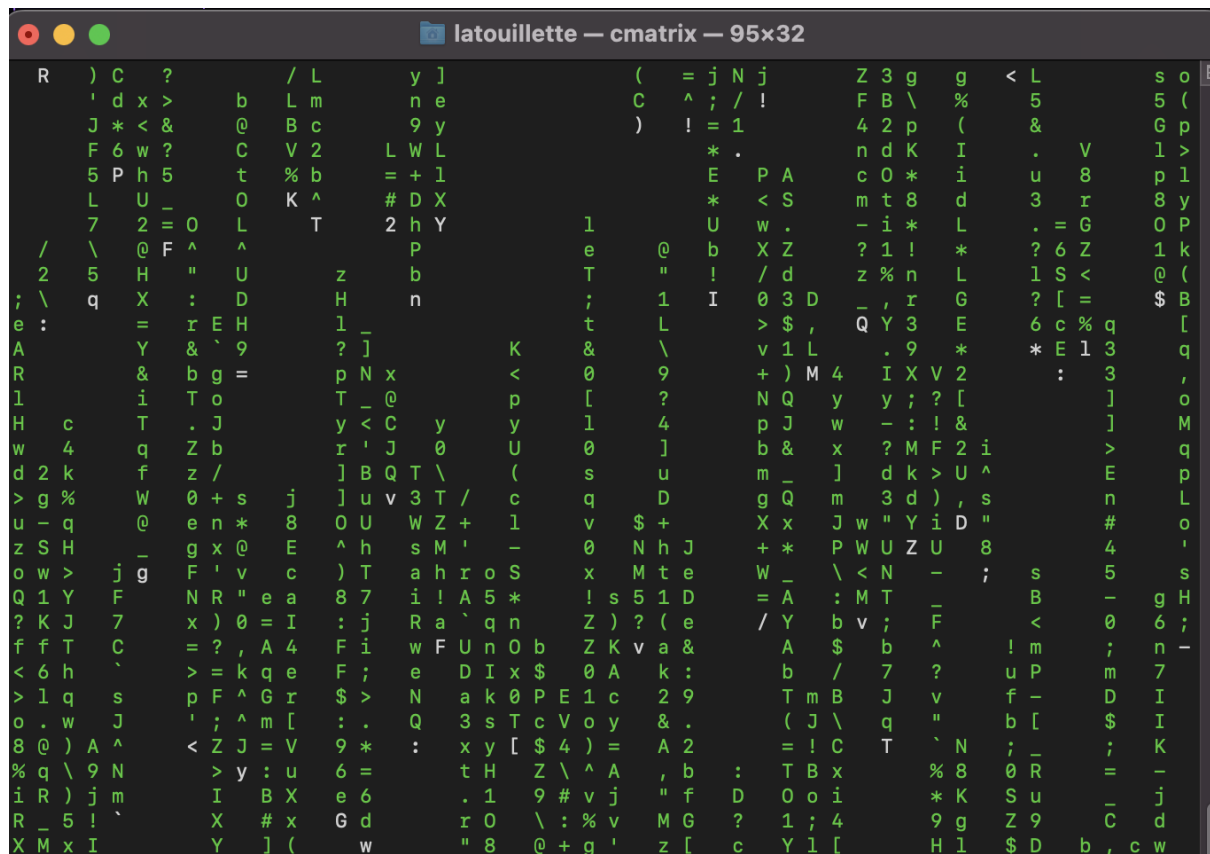


### JOB 3 :

Étant sur Mac OS j'ai dû installer un gestionnaire de paquets car apt n'est pas présent nativement sur Mac OS. J'ai donc installé HomeBrew à l'aide de cette commande :

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

Par la suite j'ai juste eu à installer le paquet cmatrix avec la commande : `brew install cmatrix`  
Il n'y avait plus qu'à lancer cmatrix.



### JOB 4 :

**Créer un fichier users.txt qui contiendra User1 et User2 séparé par un retour à la ligne**  
il faut utiliser `touch users.txt` et après éditer le fichier avec `open users.txt`

Étant encore une fois sur Mac Os, j'utilise la commande `dscl` au lieu de `Useradd` etc... car elle n'est pas disponible sur Mac Os

**Créer un groupe appelé "Plateformeurs"**

```
sudo dscl . create /Groups/Plateformeurs
```

### Créer un utilisateur appelé "User1"

```
sudo dscl . create /Users/User1
```

### Créer un utilisateur appelé "User2"

```
sudo dscl . create /Users/User2
```

### Ajouter "User2" au groupe Plateformeurs

```
append /Groups/Plateformeurs GroupMembership User2
```

J'ai vérifié que ça marchait en redémarrant mon pc et les deux Users étaient apparu au démarrage.

### Copier votre "users.txt" dans un fichier "droits.txt"

**touch droits.txt** (va créer le fichier)

**cp users.txt droits.txt** (copie entrée 1 dans entrée 2)

### Copier votre "users.txt" dans un fichier "groupes.txt"

**touch groupes.txt** (va créer le fichier)

**cp users.txt groupes.txt** (copie entrée 1 dans entrée 2)

### Changer le propriétaire du fichier "droits.txt" pour mettre "User1"

Pour utiliser chown j'ai découvert que je devais ajouter un ID(en chiffres) à mon user avec :

**sudo dscl . -create /Users/User1 UniqueID 11**

ensuite j'utilise **chown 11 droits.txt** dans le bon répertoire

```
MacBook-Air-de-Antonin:Shell latouillette$ ls -la
total 24
drwxr-xr-x  5 latouillette  staff  160  3 oct 11:46 .
drwx-----+ 11 latouillette  staff  352  3 oct 11:23 ..
-rw-r--r--@  1 11          staff   11  3 oct 11:45 droits.txt
-rw-r--r--@  1 latouillette  staff   11  3 oct 11:46 groupes.txt
-rw-r--r--@  1 latouillette  staff   11  3 oct 11:38 users.txt
MacBook-Air-de-Antonin:Shell latouillette$
```

J'ai vérifié que ça marchait en voulant modifier le fichiers droits.txt et j'ai eu un message d'erreur.

### Changer les droits du fichier "droits.txt" pour que "User2" ai accès seulement en lecture

Il suffit de changer les droits du triplet users (u) en retirant l'écriture (u-w). On voit bien que sur le fichier droits.txt seul le r apparaît sur le triplet user contrairement au screenshot précédent.

```
[→ Shell sudo chmod u-w droits.txt
[→ Shell ls -la
total 72
drwxr-xr-x  11 latouillette  staff  352  5 oct 12:21 .
drwx-----+ 11 latouillette  staff  352  3 oct 11:23 ..
-rw-r--r--@  1 latouillette  staff 6148  5 oct 12:21 .DS_Store
-r----x-w-@  1 latouillette  staff   11  3 oct 11:45 droits.txt
-r--rw-r--@  1 latouillette  staff   11  3 oct 11:46 groupes.txt
-rw-r--r--  1 latouillette  staff   11  4 oct 10:58 nb_lignes.txt
drwxr-xr-x  2 latouillette  staff   64  5 oct 12:21 scp
-rw-r--r--  1 latouillette  staff   11  4 oct 11:10 sources.save.txt
-rw-r--r--  1 latouillette  staff   11  4 oct 11:11 sources_save.txt
-rw-r--r--  1 latouillette  staff   28  4 oct 10:29 une_commande.txt
-rw-r--r--@  1 latouillette  staff   11  3 oct 11:38 users.txt
```

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

il suffit d'enlever les droits w (write) déjà existants avec **sudo chmod u-w groupes.txt**

```
[MacBook-Air-de-Antonin:Shell latouillette$ ls -la
total 24
drwxr-xr-x  5 latouillette  staff  160  3 oct 11:46 .
drwx-----+ 11 latouillette  staff  352  3 oct 11:23 ..
-r----x-w-   1 11          staff   11  3 oct 11:45 droits.txt
-r--r--r--@  1 latouillette  staff   11  3 oct 11:46 groupes.txt
-rw-r--r--@  1 latouillette  staff   11  3 oct 11:38 users.txt
MacBook-Air-de-Antonin:Shell latouillette$
```

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

Il suffit d'ajouter la permission w (write) au triplet groupe avec **chmod g+w groupes.txt**

```
[MacBook-Air-de-Antonin:Shell latouillette$ ls -la
total 24
drwxr-xr-x  5 latouillette  staff  160  3 oct 11:46 .
drwx-----+ 11 latouillette  staff  352  3 oct 11:23 ..
-r----x-w-   1 11          staff   11  3 oct 11:45 droits.txt
-r--rw-r--@  1 latouillette  staff   11  3 oct 11:46 groupes.txt
-rw-r--r--@  1 latouillette  staff   11  3 oct 11:38 users.txt
MacBook-Air-de-Antonin:Shell latouillette$
```

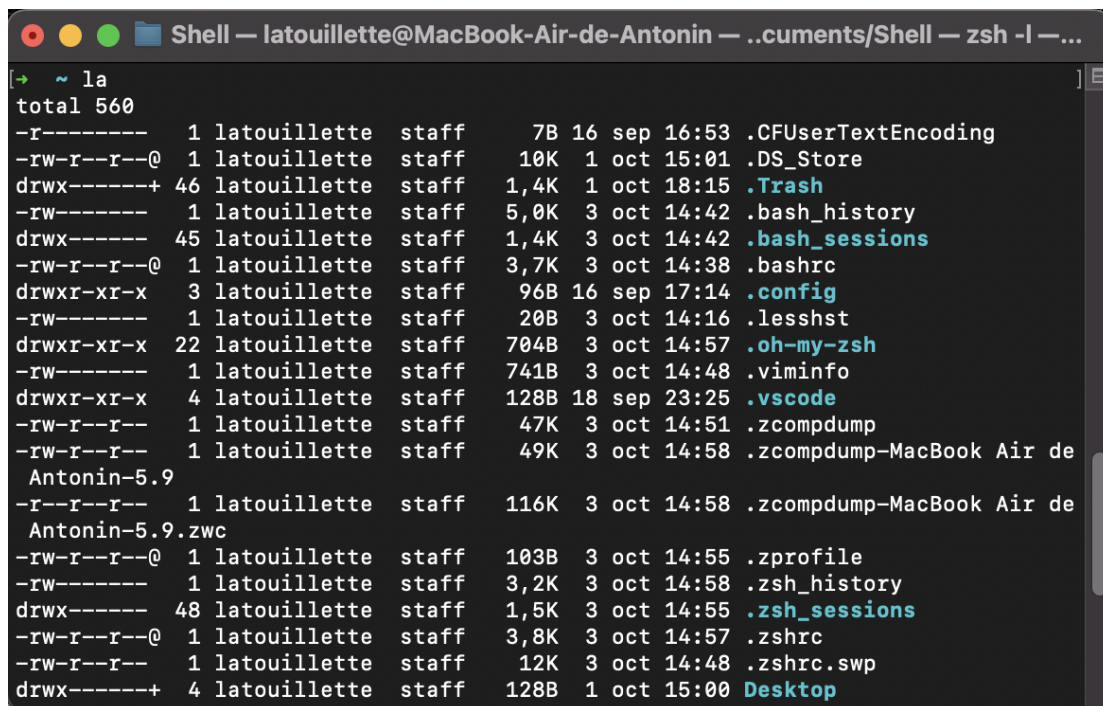
JOB 5 :

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

Mac Os ayant comme shell par défaut zsh j'ai modifié .zshrc au lieu de .bashrc mais ça revient au même

Pour ajouter l'alias la j'ai juste tapé **alias la="ls -la"** à la fin de mon fichier .zshrc

Ne pas oublier de faire **source ~/.zshrc** pour actualiser le terminal et que l'alias soit pris en compte.



```
[~] ~ la
total 560
-r----- 1 latouillette staff 7B 16 sep 16:53 .CFUserTextEncoding
-rw-r--r-- 1 latouillette staff 10K 1 oct 15:01 .DS_Store
drwx-----+ 46 latouillette staff 1,4K 1 oct 18:15 .Trash
-rw----- 1 latouillette staff 5,0K 3 oct 14:42 .bash_history
drwx----- 45 latouillette staff 1,4K 3 oct 14:42 .bash_sessions
-rw-r--r-- 1 latouillette staff 3,7K 3 oct 14:38 .bashrc
drwxr-xr-x 3 latouillette staff 96B 16 sep 17:14 .config
-rw----- 1 latouillette staff 20B 3 oct 14:16 .lessshst
drwxr-xr-x 22 latouillette staff 704B 3 oct 14:57 .oh-my-zsh
-rw----- 1 latouillette staff 741B 3 oct 14:48 .viminfo
drwxr-xr-x 4 latouillette staff 128B 18 sep 23:25 .vscode
-rw-r--r-- 1 latouillette staff 47K 3 oct 14:51 .zcompdump
-rw-r--r-- 1 latouillette staff 49K 3 oct 14:58 .zcompdump-MacBook Air de
Antonin-5.9
-r--r--r-- 1 latouillette staff 116K 3 oct 14:58 .zcompdump-MacBook Air de
Antonin-5.9.zwc
-rw-r--r-- 1 latouillette staff 103B 3 oct 14:55 .zprofile
-rw----- 1 latouillette staff 3,2K 3 oct 14:58 .zsh_history
drwx----- 48 latouillette staff 1,5K 3 oct 14:55 .zsh_sessions
-rw-r--r-- 1 latouillette staff 3,8K 3 oct 14:57 .zshrc
-rw-r--r-- 1 latouillette staff 12K 3 oct 14:48 .zshrc.swp
drwx-----+ 4 latouillette staff 128B 1 oct 15:00 Desktop
```

**Ajouter un alias qui permettra de lancer la commande "apt-get update" en tapant "update"**

Ayant installé comme gestionnaire de paquets Homebrew ma commande d'update est brew update, je vais donc lui créer un alias avec **alias update="brew update"**

**Ajouter un alias qui permettra de lancer la commande "apt-get upgrade" en tapant "upgrade"**

Ayant installé comme gestionnaire de paquets Homebrew ma commande d'update est brew update, je vais donc lui créer un alias avec **alias upgrade="brew upgrade"**

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

Il suffit d'ajouter après mes alias une ligne : **USER='latouillettez'** dans le .zshrc

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

il faut utiliser **source ~/.zshrc** qui actualise le shell

**Afficher les variables d'environnement**

il faut utiliser **printenv** qui affiche toutes les variables d'environnement.

on voit bien que ma variable USER a bien été prise en compte, mais on peut aussi le vérifier avec **echo \$USER** (le dollar demande à ce que ça soit la valeur de USER qui soit affichée).



```
latouillette — latouillette@MacBook-Air-de-Antonin — ~ — -zsh — 80x24
[~] ~ printenv
__CFBundleIdentifier=com.apple.Terminal
TMPDIR=/var/folders/tj/6mtkh69d5sq5c1xbw3n487700000gn/T/
XPC_FLAGS=0x0
TERM=xterm-256color
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.nTasEEJVsi/Listeners
XPC_SERVICE_NAME=0
TERM_PROGRAM=Apple_Terminal
TERM_PROGRAM_VERSION=445
TERM_SESSION_ID=167956B5-5B04-49ED-9C9B-54C52B27C0A4
SHELL=/bin/zsh
HOME=/Users/latouillette
LOGNAME=latouillette
USER=latouillettez
PATH=/opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
SHLVL=1
PWD=/Users/latouillette
OLDPWD=/Users/latouillette
HOMEBREW_PREFIX=/opt/homebrew
HOMEBREW_CELLAR=/opt/homebrew/Cellar
HOMEBREW_REPOSITORY=/opt/homebrew
MANPATH=/opt/homebrew/share/man::
INFOPATH=/opt/homebrew/share/info:
```

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

Pour le faire de manière temporaire (le PATH se remettra par défaut à la déconnexion du terminal) on peut faire :

```
→ Downloads export PATH=$PATH:/home/$USER/Bureau
→ Downloads echo $PATH
/opt/homebrew/bin:/opt/homebrew/sbin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/home/latouillettez/Bureau
→ Downloads
```

on voit bien avec **echo \$PATH** que le path a bien été changé.

Si on voulait le faire définitivement il faudrait aller dans le `.zshrc` et changer manuellement le PATH.

JOB 6:

**tar -xvf GhostintheShell.tar.gz** désarchive le fichier GhostintheShell.tar.gz dans le répertoire courant. (-xvf sont des options qui servent à afficher en sortie le fichier résultant de l'opération (v), extraire l'archive (x) et utiliser le fichier en paramètre (f)).

```
→ Downloads tar -xvf GhostintheShell.tar.gz
x Ghost in the Shell.pdf
→ Downloads
```

JOB 7:

**Créer un fichier “une\_commande.txt” avec le texte suivant “Je suis votre fichier texte”**

j'utilise **echo "Je suis votre fichier texte" > une\_commande.txt** le > va rediriger la sortie d'echo (qui affiche le texte donné en option) et l'écrire dans le fichier texte une\_commande.txt qui sera créée en même temps (si il ne l'est pas déjà et dans ce cas là on peut utiliser >> pour concaténer et ne pas écraser le fichier).

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

**wc -l etc/apt/sources.list > nb\_lignes.txt**

N'ayant pas trouvé d'équivalent de fichier sources.list pour homebrew je fait la commande avec apt. wc est une commande qui compte le nombre de lignes, de mots, le nombre d'octets et qui retourne le nom du fichier en paramètre. L'option -l permet de ne récupérer que le nombre de ligne en sortie et on redirige vers nb\_lignes.txt

**Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé “save\_sources”**

**cat etc/apt/sources.list && cat etc/apt/sources.list > save\_sources.txt**

on affiche le contenu du fichier et on utilise && pour faire la même commande et la rediriger vers notre fichier. j'ai fait ça car juste **cat etc/apt/sources.list > save\_sources.txt** n'affiche pas la sortie cat dans le terminal.

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

on utilise **find .\* && grep "alias"** qui, avec find cherche tous les fichiers commençant par . (avec le \* qui signifie qu'il peut y avoir n'importe quoi APRES le point) dans le répertoire courant. Ensuite on utilise && qui permet de garder la première condition et d'ajouter la deuxième (un ET logique) : les fichiers qui contiennent “alias”.

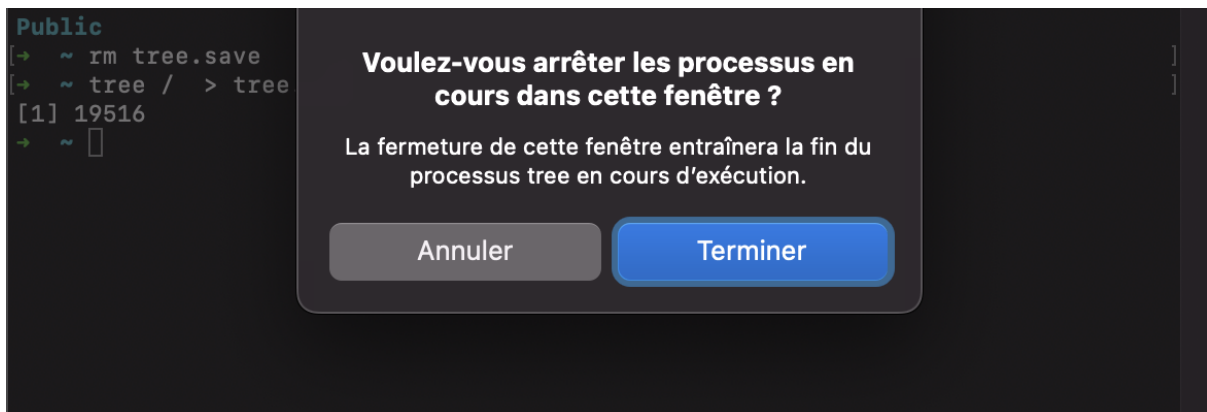
Pour aller plus loin :

**Installer la commande tree**

j'installe la commande tree avec **brew install tree**

**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”**

j'utilise **tree / > tree.save &** qui redirige la sortie vers tree.save. Pour le faire passer en arrière plan je mets à la fin de ma commande & ce qui lance le programme en arrière plan. C'est vérifiable facilement car quand j'essaye de fermer mon terminal j'ai un message me prévenant qu'un processus est en cours.



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

**ls -l | wc -l**

j'utilise **ls -l** pour lister tous les fichiers du répertoire et j'utilise sa sortie avec **wc -l** qui va compter ces éléments.

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

l'update et l'upgrade se sont bien passés.

BONUS :

**Installer SSH**

sur mac OS ssh est installé nativement mais pas toujours activé. Pour vérifier qu'il est activé je tape **sudo systemsetup -getremotelogin** . J'ai eu en sortie **Remote Login : Off** ce qui signifie qu'il est désactivé. Pour l'activer je tape **sudo systemsetup -setremotelogin on** .

**Générer une clé SSH**

Pour ce faire je crée un dossier ssh avec **mkdir ssh** dans le répertoire home

je me déplace dans ce dossier et je génère une clé avec **ssh-keygen**. Je renseigne le dossier souhaité, ma passphrase et le script va créer un fichier ssh qui contient ma clé privée à ne surtout pas montrer et un fichier ssh.pub qui contient ma clé publique partageable.

```
ssh — latouillette@MacBook-Air-de-Antonin — ~/ssh — zsh — 93x29
→ ssh ssh keygen
ssh: Could not resolve hostname keygen: nodename nor servname provided, or not known
→ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/latouillette/.ssh/id_rsa): ./ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./ssh
Your public key has been saved in ./ssh.pub
The key fingerprint is:
SHA256:aAQ3s4QkFfGctGgdkJ92++koY04xNCmDupfgolZntic latouillette@MacBook-Air-de-Antonin.local
The key's randomart image is:
+---[RSA 3072]-----+
| .oB*B |
| ooX O |
| . =.% |
| . . =+O |
| o . =.S |
|.o o = o |
|o.+ + o . |
|oo E+. .o |
| o o+o... |
+---[SHA256]-----+
→ ssh ls
ssh      ssh.pub
```

### Se connecter à une VM ou l'ordinateur d'un camarade via SSH

Avant de me connecter en SSH j'ai dû ouvrir mon port 22 qui était fermé par défaut. Pour se connecter à un autre ordinateur en SSH il faut faire `ssh utilisateur@ip_host` où l'utilisateur est le nom d'utilisateur de l'ordinateur auquel on veut accéder et `ip_host` son adresse ip. Il faut connaître le mot de passe de son camarade et après on a bien accès à son arborescence.

### Configurer SSH pour empêcher le login root (root ne peut pas se connecter en SSH)

Pour enlever l'accès SSH au root il faut aller modifier la ligne `PermitRootLogin` dans le fichier `sshd_config` présent dans `/etc/ssh/`.

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

### Modifier le port de connexion de SSH (autre que 22)

Pour modifier le port il faut aussi aller dans `sshd_config` et modifier la ligne correspondant au port.

### Ensuite se connecter en SSH sans avoir à renseigner de mot de passe

Pour ne pas avoir à renseigner de mot de passe on utilise `ssh-copy-id -i ssh.pub user@ip_host` avec nos adresses mutuelles pour se partager nos clés publiques.

### Uploader un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade)

Pour uploader un fichier sur le pc de l'autre il faut utiliser le protocole `scp` avec la commande **scp** `chemin_du_fichier user@ip_host:chemin_d'arrivée`. Elle agit comme la commande `cp`.

### Télécharger un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade)

Pour télécharger un fichier il faut utiliser `scp` mais à l'inverse. C'est à dire **scp** `user@ip_host:chemin_du_fichier_voulu chemin_d'arrivée`

**Limiter l'utilisation de SSH à un groupe particulier nommé "Plateforme\_ssh"**

Pour limiter l'accès à un groupe il faut encore aller dans sshd\_config et ajouter une ligne **AllowGroups Plateforme\_ssh latouillette**

**Quel est l'intérêt d'utiliser SSH ?**

L'intérêt d'utiliser SSH c'est qu'il permet un transfert de fichiers sécurisé grâce à un protocole de cryptage. Il permet aussi aux utilisateurs d'exécuter des commandes shell sur des machines distantes.

**Est-ce que les clés générées par SSH par défaut sont assez sécurisées ? Justifier votre réponse**

Je suppose que non car il y a des options qui vont avec ssh-keygen qui permette d'utiliser plusieurs protocoles de cryptage (-b par exemple) et on peut choisir sur combien de bits générer la clé (4096bits par exemple).

**Citez d'autres protocoles de transfert ? Quelles sont les différences entre ses protocoles ?**

Il y a FTP qui utilise une transmission via deux canaux mais qui n'utilise pas de chiffrement.

Il y a HTTP qui sert pour le web et définit le format des messages envoyés entre les navigateurs et les serveurs.