

~\$hell

Abraham UKACHI

9 Octobre 2022

# Sommaire

<b>Introduction</b>	<b>5</b>		
<b>macOS</b>	<b>7</b>	<b>Job 06</b>	<b>24</b>
<b>Job 01</b>	<b>8</b>	6.1 Télécharger l'archive	
1.1 Afficher le manuel de la		"ghost_in_the_shell.tar.gz"	24
commande ls	8	6.2 Désarchiver le fichier	24
1.2 Afficher les fichiers cachés du		<b>Job 07</b>	<b>25</b>
home de votre utilisateur	9	7.1 Créer un fichier	
1.3 Afficher les fichiers cachés plus		"une_commande.txt" avec le texte	
les informations sur les droits sous		suivant "Je suis votre fichier texte"	
forme de liste	9	25	
<b>Job 02</b>	<b>11</b>	7.2 Compter le nombre de lignes	
2.1 Afficher les 10 premières		présentes dans votre fichier de	
lignes du fichier ".bashrc"	11	source apt et les enregistrer dans	
2.2 Afficher les 10 dernières lignes		un fichier nommé "nb_lignes.txt" 25	
du fichier ".bashrc"	11	7.3 Afficher le contenu du fichier	
2.3 Afficher les 20 premières lignes		source apt et l'enregistrer dans un	
du fichier ".bashrc"	12	autre fichier appelé	
2.4 Afficher les 20 dernières lignes		"save_sources"	26
du fichier ".bashrc"	12	7.4 Faites une recherche des	
<b>Job 03</b>	<b>13</b>	fichiers commençant par "." tout en	
3.1 Installer le paquet "cmatrix"	13	cherchant le mot alias qui sera	
3.2 lancer le paquet que vous		utilisé depuis un fichier	26
venez d'installer	13	<b>Pour aller plus loin</b>	<b>27</b>
3.3 Mettre à jour son gestionnaire		8.1 Installer la commande tree	27
de paquets	13	8.2 Lancer la commande tree en	
3.4 Mettre à jour ses différents		arrière-plan qui aura pour but	
logiciels	14	d'afficher toute l'arborescence en	
3.5 Télécharger les internets :		de votre / en enregistrant le	
Google	14	résultat dans un fichier "tree.save"	
3.6 Redémarrer votre machine	14	27	
3.7 Éteindre votre machine	14	8.3 lister les éléments présents	
<b>Job 04</b>	<b>16</b>	dans le dossier courant est utilisé	
4.1 Créer un fichier users.txt qui		directement le résultat de votre	
contiendra User1 et User2 séparé		première commande pour compter	
par un retour à la ligne	16	le nombre d'éléments trouvés	28
4.2 Créer un groupe appelé		8.4 Lancer une commande pour	
"Plateformeurs"	16	update vos paquets, si l'update	
4.3 Créer un utilisateur appelé		réussi alors, vous devrez lancer un	
"User1"	17	upgrade de vos paquets. Si	
4.4 Créer un utilisateur appelé		l'update échoue, votre upgrade ne	
"User2"	17	se lancera pas	28
		<b>Questions</b>	<b>29</b>

4.5 Ajouter "User2" au groupe Plateformeurs	18	Comment ajouter des options à une commande ?	29
4.6 Copier votre "users.txt" dans un fichier "droits.txt"	18	Quelles sont les deux syntaxes principales d'écriture des options pour une commande ?	29
4.7 Copier votre "users.txt" dans un fichier "groupes.txt"	18		
4.8 Changer le propriétaire du fichier "droits.txt" pour mettre "User1"	19	Draft	
4.9 Changer les droits du fichier "droits.txt" pour que "User2" ai accès seulement en lecture	19		
4.10 Changer les droits du fichier "groupes.txt" pour que les utilisateurs puissent accéder au fichier en lecture uniquement	19		
4.11 Changer les droits du fichier pour que le groupe "Plateformeurs" puissent y accéder en lecture/écriture.	20		
<b>Job 05</b>	<b>21</b>		
5.1 Ajouter un alias qui permettra de lancer la commande "ls -la" en tapant "la"	21		
5.2 Ajouter un alias qui permettra de lancer la commande "apt-get update" en tapant "update"	21		
5.3 Ajouter un alias qui permettra de lancer la commande "apt-get upgrade" en tapant "upgrade"	21		
5.4 Ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à votre nom d'utilisateur	22		
5.5 Mettre à jour les modifications de votre bashrc dans votre shell actuel	22		
5.6 Afficher les variables d'environnement	22		
5.7 Ajouter à votre Path le chemin "/home/'votre utilisateur'/Bureau"	23		

# Introduction

> **IMPORTANT:** Cette documentation est un travail en cours et va bientôt changer.

**C'est quoi un shell?** Tout simplement, un shell est un programme qui fait office d'interface entre vous et l'ordinateur sous la forme d'un interpréteur de commandes. En gros, c'est ce que vous voyez quand vous lancez un terminal. Il en existe plusieurs comme : **ash** (Almquist shell), **sh** (Bourne shell), **bash** (Bourne Again Shell), **csh** (C shell), **zsh** (Z shell), **ksh** (Korn shell), etc.

Dans cette documentation, nous allons nous focaliser sur le "Z shell" et exécuter des commandes **UNIX** sur mac OS, malgré le fait que les "jobs" ci-dessous précisent les commandes Linux. Je n'ai rien contre les autres shells et/ou systèmes d'exploitations, au contraire j'avais prévu ajouté 2 titres d'OS (**Manjaro** & **Debian**) pour expliquer d'avantages les différentes lignes de command utilisées sur ces systèmes d'exploitation, mais j'ai été pris par le temps et devait rendre ce projet. En plus, j'utilise un **MacBook Air** au quotidien sur lequel j'aimerais plus me familiariser avec l'interpréteur de commandes et/ou langage de programmation. Cela dit, il est encore possible qu' à l'avenir, je reviennent sur cette documentation pour la mettre à jour 😊.

Cette documentation a été faite de telle sorte que même les débutants puissent suivre / comprendre, par conséquent certaines explications ont été délibérément répétées. Les lignes de commandes ont été colorisées pour cette raison et également pour des raisons visuelles. Voici ce que ces couleurs signifient:

- **Cyan foncé 1** : Le numéro de la ligne de commande;
- **Rouge foncé 2** : Une commande qui sert à exécuter une commande en tant qu'un autre utilisateur sous l'autorisation du fichier de configuration /etc/sudoers;
- **Jaune foncé 3** : Une commande disponible sur l'OS;
- **Gris foncé 4** : Options, arguments, textes ou autres.

P.S. La réponse aux [questions](#) posées dans les consignes se trouve au bout de cette documentation.



# macOS

version 10.14.6 ( **Mojave** )

# Job 01



JOB: Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

## 1.1 Afficher le manuel de la commande ls

Vous pouvez afficher le manuel de la commande `ls` avec cette commande:

```
man ls
```

- `man` - (*commande*) permet de visionner le manuel d'une commande ou le manuel d'un fichier de configuration.
- `ls` - (*commande*) répertorie les fichiers et les répertoires dans le système de fichiers et affiche des informations détaillées sur eux.

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

NAME
  ls -- list directory contents

SYNOPSIS
  ls [-ABCFGHLOPRSTUW@abcdefghijklmnopqrstuvwxyz] [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, asso-
  ciated information.

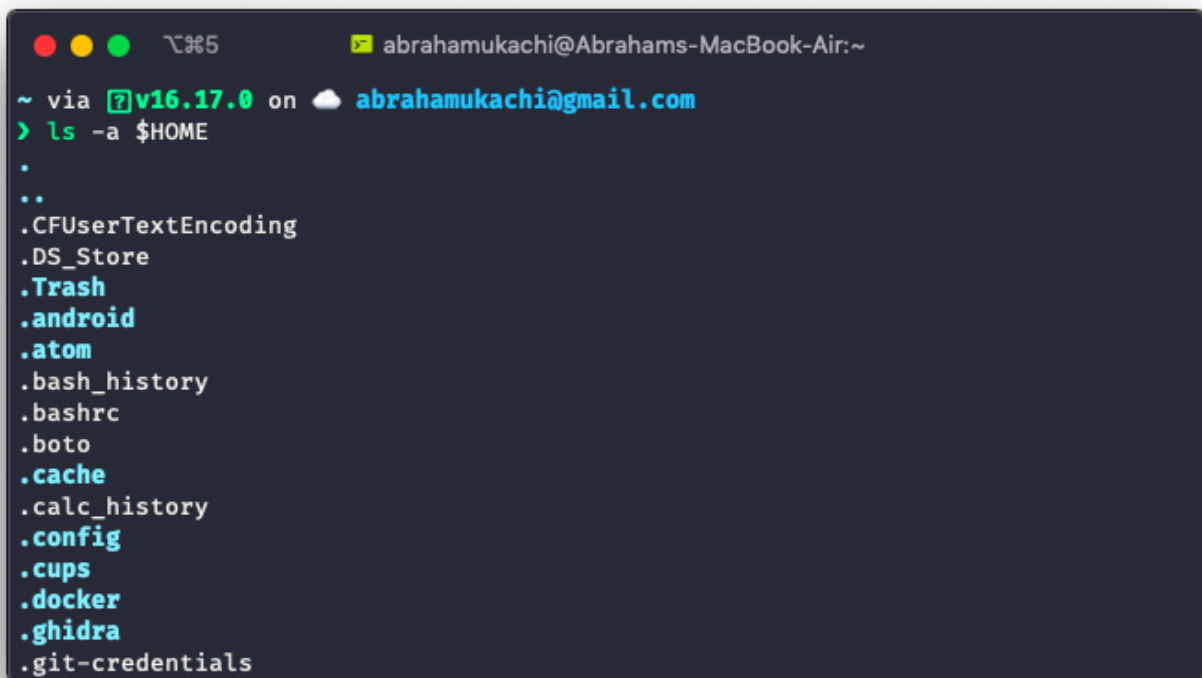
  If no operands are given, the contents of the current directory are dis-
  played. If more than one operand is given, non-directory operands are
```

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

Vous pouvez afficher les fichiers cachés du home de votre utilisateur à l'aide de la commande suivante:

```
ls -a $HOME
```

- ``ls`` - (*commande*) répertorie les fichiers et les répertoires dans le système de fichiers et affiche des informations détaillées sur eux.
- ``-a`` - (*option*) affiche les fichiers cachés (ceux commençant par un point).
- ``$HOME`` - (*variable/chemin*) c'est un [alias](#) du home.



```

~ via [?] v16.17.0 on [?] abrahamukachi@gmail.com
> ls -a $HOME
.
..
.CFUserTextEncoding
.DS_Store
.Trash
.android
.atom
.bash_history
.bashrc
.boto
.cache
.calc_history
.config
.cups
.docker
.ghidra
.git-credentials

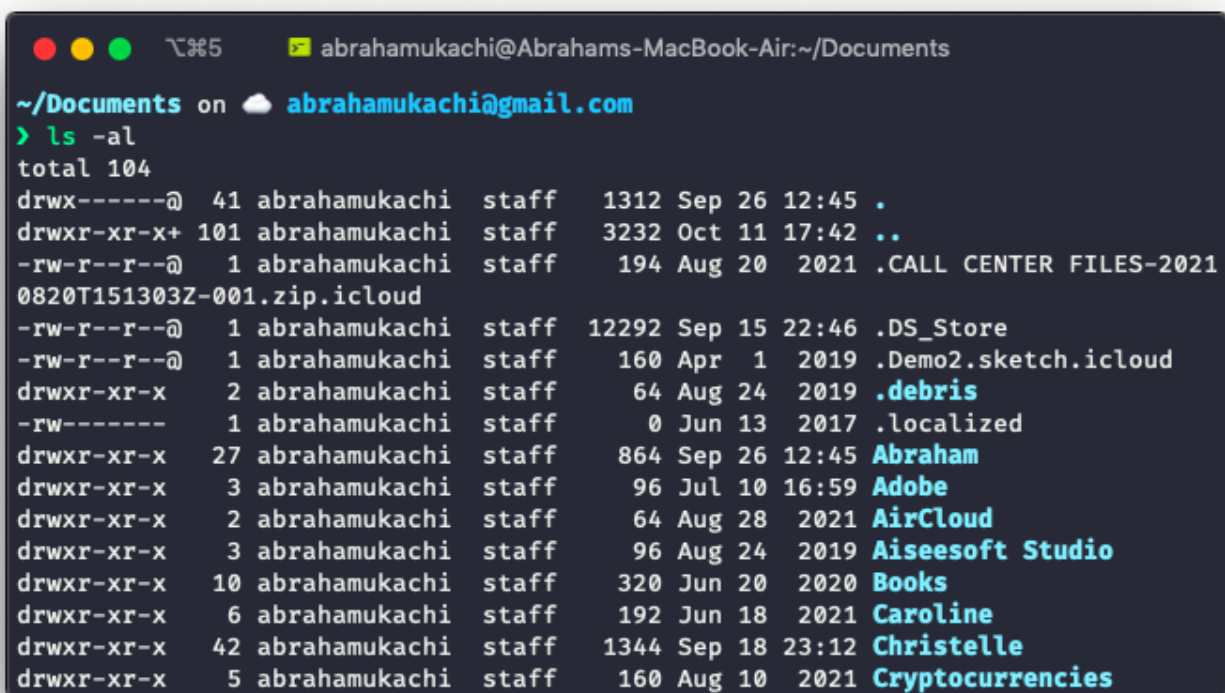
```

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

Vous pouvez afficher des fichiers cachés plus les informations sur les droits sous forme de liste sur Linux / UNIX à l'aide de la commande suivante:

```
ls -al
```

- ``ls`` - (*commande*) répertorie les fichiers et les répertoires dans le système de fichiers et affiche des informations détaillées sur eux.
- ``-al`` - (*options*) l'option ``a`` affiche les fichiers cachés (ceux commençant par un point) et l'option ``l`` affiche le format de liste longue



```

~/Documents on iCloud abrahamukachi@gmail.com
> ls -al
total 104
drwx-----@ 41 abrahamukachi  staff   1312 Sep 26 12:45 .
drwxr-xr-x+ 101 abrahamukachi  staff   3232 Oct 11 17:42 ..
-rw-r--r--@  1 abrahamukachi  staff    194 Aug 20  2021 .CALL CENTER FILES-2021
0820T151303Z-001.zip.icloud
-rw-r--r--@  1 abrahamukachi  staff  12292 Sep 15 22:46 .DS_Store
-rw-r--r--@  1 abrahamukachi  staff    160 Apr  1  2019 .Demo2.sketch.icloud
drwxr-xr-x  2 abrahamukachi  staff     64 Aug 24  2019 .debris
-rw-----  1 abrahamukachi  staff      0 Jun 13  2017 .localized
drwxr-xr-x 27 abrahamukachi  staff    864 Sep 26 12:45 Abraham
drwxr-xr-x  3 abrahamukachi  staff     96 Jul 10 16:59 Adobe
drwxr-xr-x  2 abrahamukachi  staff     64 Aug 28  2021 AirCloud
drwxr-xr-x  3 abrahamukachi  staff     96 Aug 24  2019 Aiseesoft Studio
drwxr-xr-x 10 abrahamukachi  staff    320 Jun 20  2020 Books
drwxr-xr-x  6 abrahamukachi  staff    192 Jun 18  2021 Caroline
drwxr-xr-x 42 abrahamukachi  staff   1344 Sep 18 23:12 Christelle
drwxr-xr-x  5 abrahamukachi  staff    160 Aug 10  2021 Cryptocurrencies

```



# Job 02



JOB: Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

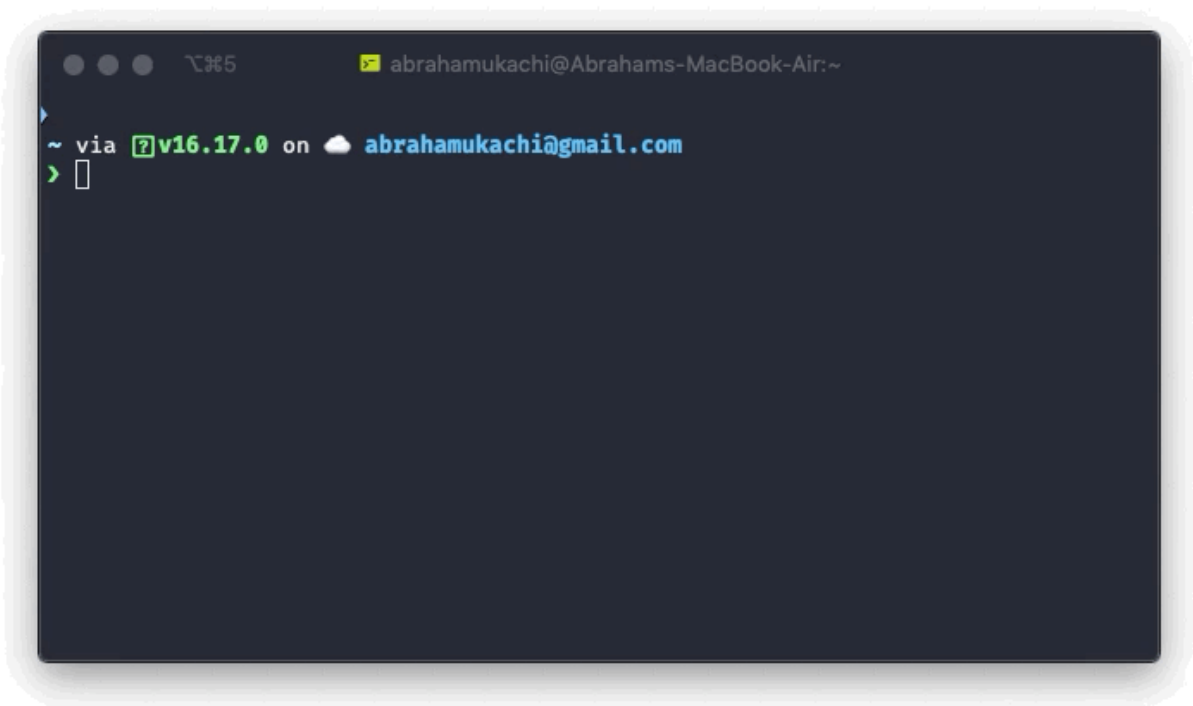
## 2.1 Lisez un fichier en utilisant une commande qui permet seulement de lire

Vous pouvez lire un fichier sur Linux/UNIX avec la commande suivante:

```
cat feeling.txt
```

- `cat` - (*commande*) permet de lire un fichier.
- `feeling.txt` - (*fichier*) le nom du fichier à lire.

Pour se placer dans le bon répertoire, on utilise la commande `cd` puis on navigue dans le répertoire. Voir le résultat ci dessous:



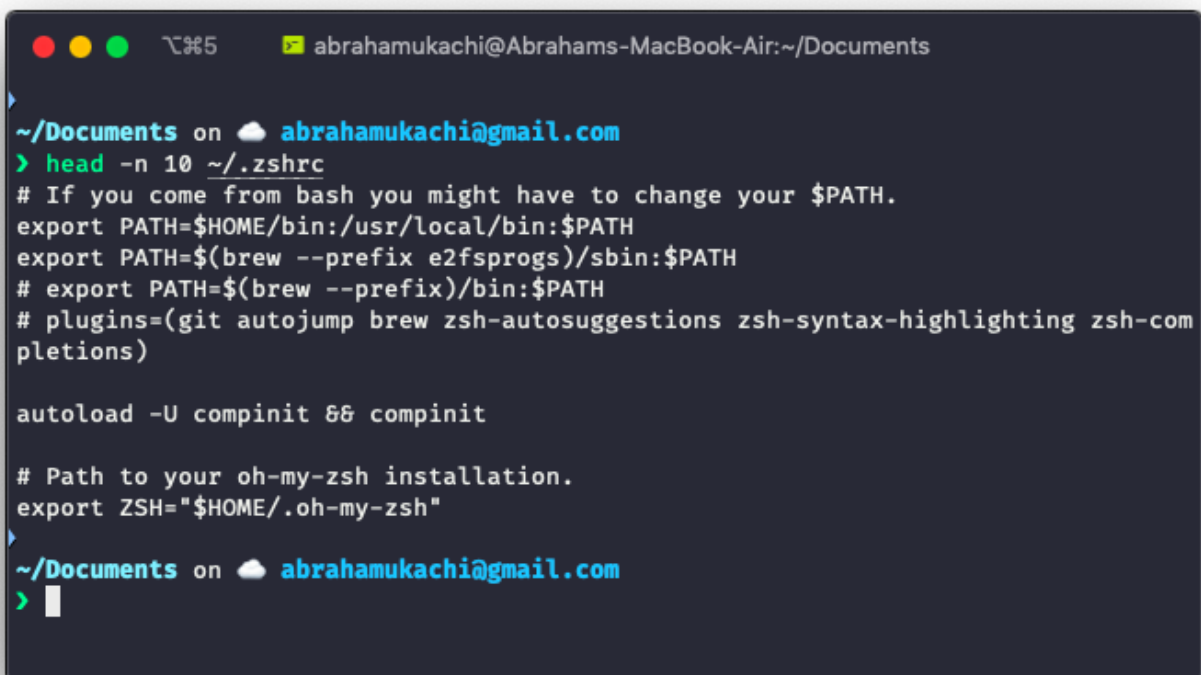
## 2.2 Afficher les 10 premières lignes du fichier “.zshrc”

Le fichier “.zshrc” a été utilisé au lieu de “.bashrc” parce que le shell installé/utilisé sur cet OS est ZSH et non BASH.

Vous pouvez afficher les 10 premières lignes du fichier “.zshrc” à l’aide de la commande suivante:

```
head -n 10 ~/.zshrc
```

- `head` - (*commande*) affiche le début d’un fichier texte ou limite l’affichage à une certaine zone.
- `-n 10` - (*option + argument*) l’option `-n` avec le nombre qui le suit, on détermine le nombre de lignes à afficher, alors les **10** premières lignes d’un fichier.
- `~/.zshrc` - (*fichier*) un fichier texte caché ``.`` dans le home `~` de l’utilisateur, et utilisé par la command `head`



```

~/Documents on ⬤ abrahamukachi@gmail.com
> head -n 10 ~/.zshrc
# If you come from bash you might have to change your $PATH.
export PATH=$HOME/bin:/usr/local/bin:$PATH
export PATH=$(brew --prefix e2fsprogs)/sbin:$PATH
# export PATH=$(brew --prefix)/bin:$PATH
# plugins=(git autojump brew zsh-autosuggestions zsh-syntax-highlighting zsh-com
pletions)

autoload -U compinit && compinit

# Path to your oh-my-zsh installation.
export ZSH="$HOME/.oh-my-zsh"
~/Documents on ⬤ abrahamukachi@gmail.com
>

```

## 2.3 Afficher les 10 dernières lignes du fichier “.bashrc”

Le fichier “.zshrc” a été utilisé au lieu de “.bashrc” parce que le shell installé/utilisé sur cet OS est ZSH et non BASH.

Vous pouvez afficher les 10 dernières lignes du fichier “.zshrc” à l’aide de la commande suivante:

```
tail -n 10 ~/.zshrc
```

- ``tail`` - (*commande*) affiche la fin d’un fichier texte ou limite l’affichage à une certaine zone de manière analogue à la commande ``head``.
- ``-n 10`` - (*option + argument*) l’option ``-n`` avec le nombre qui le suit, on détermine le nombre de lignes à afficher, alors les **10** dernières lignes d’un fichier.
- ``~/.zshrc`` - (*fichier*) un fichier texte caché ``.`` dans le home ``~`` de l’utilisateur, et utilisé par la command ``tail``

## 2.4 Afficher les 20 premières lignes du fichier “.bashrc”

Le fichier “.zshrc” a été utilisé au lieu de “.bashrc” parce que le shell installé/utilisé sur cet OS est ZSH et non BASH.

Vous pouvez afficher les 20 premières lignes du fichier “.zshrc” à l’aide de la commande suivante:

```
head -n 20 ~/.zshrc
```

- ``head`` - (*commande*) affiche le début d’un fichier texte ou limite l’affichage à une certaine zone.
- ``-n 20`` - (*option + argument*) l’option ``-n`` avec le nombre qui le suit, on détermine le nombre de lignes à afficher, alors les **20** premières lignes d’un fichier.
- ``~/.zshrc`` - (*fichier*) un fichier texte caché ``.`` dans le home ``~`` de l’utilisateur, et utilisé par la command ``head``

## 2.5 Afficher les 20 dernières lignes du fichier “.bashrc”

Le fichier “.zshrc” a été utilisé au lieu de “.bashrc” parce que le shell installé/utilisé sur cet OS est ZSH et non BASH.

Vous pouvez afficher les 20 dernière lignes du fichier “.zshrc” à l’aide de la commande suivante:

```
tail -n 20 ~/.zshrc
```

- `tail` - (*commande*) affiche la fin d’un fichier texte ou limite l’affichage à une certaine zone de manière analogue à la commande `head`.
- `-n 20` - (*option + argument*) l’option `-n` avec le nombre qui le suit, on détermine le nombre de lignes à afficher, alors les **20** dernières lignes d’un fichier.
- `~/.zshrc` - (*fichier*) un fichier texte caché ``.`` dans le home `~` de l’utilisateur, et utilisé par la command `tail`

# Job 03



JOB: Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

## 3.1 Installer le paquet “cmatrix”

Vous pouvez installer le paquet **cmatrix** à l’aide de la commande suivante:

```
brew install cmatrix
```

- ``brew`` - (*commande*) gestionnaire de paquets gratuit et open-source qui s'installe sur macOS qui permet d'installer en ligne de commande des logiciels/applications.
- ``install`` - (*commande de brew*) l'alias que
- ``cmatrix`` - l'alias que

### 3.2 lancer le paquet que vous venez d'installer

Vous pouvez lancer le paquet que vous venez d'installer avec cette commande:

```
cmatrix
```

- ``cmatrix`` - l'alias que nous avons créé dans le [job 5.2](#)

### 3.3 Mettre à jour son gestionnaire de paquets

```
brew update
```

- ``brew`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``update`` - l'alias que

### 3.4 Mettre à jour ses différents logiciels

```
brew upgrade
```

- ``brew`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``upgrade`` - l'alias que

### 3.5 Télécharger les internets : Google

```
brew cask install google-chrome
```

- ``brew`` - l'alias que nous avons créé dans le [job 5.2](#)

- ``cask`` - l'alias que
- ``install`` - l'alias que
- ``google-chrome`` - l'alias que nous avons créé dans le [job 5.2](#)

### 3.6 Redémarrer votre machine

```
sudo reboot
```

- ``sudo`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``reboot`` - l'alias que

### 3.7 Éteindre votre machine

```
sudo shutdown -h now
```

- ``sudo`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``shutdown`` - l'alias que
- ``-h now`` - l'alias que

# Job 04



JOB: Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

## 4.1 Créer un fichier users.txt qui contiendra User1 et User2 séparé par un retour à la ligne

```
echo "User1\nUser2" > users.txt
```

- ``echo`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``User1\nUser2`` - l'alias que
- ``>`` - l'alias que
- ``users.txt`` - l'alias que nous avo

## 4.2 Créer un groupe appelé "Plateformeurs"

```
sudo dscl . -create /Groups/Plateformeurs PrimaryGroupID  
666
```

- ``sudo`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``dscl`` - l'alias que
- ``.`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``-create`` - l'alias que
- ``/Groups/Plateformeurs`` - l'alias que k l k lkkkojoiioj k jokjooj kjojokjo  
jkojkjokjkjo kjjok
- ``PrimaryGroupID 666`` - l'alias que

### 4.3 Créer un utilisateur appelé “User1”

```
#1 sudo dscl . -create /Users/User1 UniqueID 777
#2 sudo dscl . -create /Users/User1 PrimaryGroupID 20
```

- ``sudo`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``dscl`` - l’alias que
- ``.`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``-create`` - l’alias que
- ``/Users/User1`` - l’alias quojokjko kjjok
- ``UniqueID 777`` - l’alias que
- ``PrimaryGroup 20`` - l’alias que

### 4.4 Créer un utilisateur appelé “User2”

```
#1 sudo dscl . -create /Users/User2 UniqueID 888
#2 sudo dscl . -create /Users/User2 PrimaryGroupID 20
```

- ``sudo`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``dscl`` - l’alias que
- ``.`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``-create`` - l’alias que
- ``/Users/User2`` - l’alias quojokjko kjjok
- ``UniqueID 888`` - l’alias que
- ``PrimaryGroup 20`` - l’alias que



## 4.5 Ajouter “User2” au groupe Plateformeurs

```
sudo dscl . -append /Groups/Plateformeurs GroupMembership
User2
```

- ``sudo`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``dscl`` - l’alias que
- ``.`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``-append`` - l’alias que
- ``/Groups/Plateformeurs`` - l’alias qujokjkjo kjjok
- ``GroupMembership`` - l’alias que

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

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

- ``cp`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``users.txt`` - l’alias que
- ``droits.txt`` - l’alias que nous avons créé dans le [job 5.2](#)

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

```
cp users.txt groupes.txt
```

- ``cp`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``users.txt`` - l’alias que
- ``groupes.txt`` - l’alias que nous avons créé dans le [job 5.2](#)

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

```
sudo chown User1 droits.txt
```

- ``sudo`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``chown`` - l’alias que
- ``User1`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``droits.txt`` - l’alias que

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

```
#1 sudo chgrp Plateformeurs droits.txt
#2 sudo chmod g+r droits.txt
```

- ``sudo`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``chgrp`` - l’alias que
- ``Plateformeurs`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``droits.txt`` - l’alias que
- ``chmod`` - l’alias qujokjkjo kjjok
- ``g+r`` - l’alias que

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

```
sudo chmod o+r groupes.txt
```

- ``sudo`` - l’alias que nous avons créé dans le [job 5.2](#)

- ``chmod`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``o+r`` - l'alias que
- ``groupes.txt`` - l'alias quojokjko kjok

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

```
#1 sudo chgrp Plateformeurs groupes.txt
#2 sudo chmod g=rw groupes.txt
```

- ``sudo`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``chgrp`` - l'alias que
- ``Plateformeurs`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``groupes.txt`` - l'alias que
- ``chmod`` - l'alias quojokjko kjok
- ``g=rw`` - l'alias que
- ``groupes.txt`` - l'alias que

# Job 05



JOB: Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

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

```
alias la="ls -la"
```

- ``alias`` - une command que nous avons créé dans le [job 5.2](#)
- ``la`` - le nom de notre alias
- ``ls -la`` - l'alias que nous avons créé dans le [job 5.2](#)

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

```
alias update="brew update"
```

- ``aliase`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``update`` - l'alias que
- ``brew update`` - l'alias que nous avons créé dans le [job 5.2](#)

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

```
alias upgrade="brew upgrade"
```

- ``alias`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``upgrade`` - l'alias que
- ``brew upgrade`` - l'alias que nous avons créé dans le [job 5.2](#)

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

```
export USER="abrahamukachi"
```

- ``export`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``USER`` - l'alias que
- ``"abrahamukachi"`` - l'alias que nous avons créé dans le [job 5.2](#)

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

```
source ~/.zshrc
```

- ``source`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``~/.zshrc`` - l'alias que

#### 5.6 Afficher les variables d'environnement

```
printenv
```

- ``printenv`` - l'alias que nous avons créé dans le [job 5.2](#)

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

```
export PATH="$PATH:/home/$USER/Bureau"
```

- `export` - l'alias que nous avons créé dans le [job 5.2](#)
- `PATH` - l'alias que
- `"PATH:/home/$USER/Bureau"` - l'alias que nous avons créé dans le [job 5.2](#)

# Job 06



JOB: Vous devez télécharger l'archive suivante et la désactiver seulement avec le terminal. Cette manipulation vous permettra d'accéder à la suite du sujet.

<https://drive.google.com/file/d/1wrip6bto9ni5yyC3ej3zxDtqELazHPHz/view?usp=sharing>

## 6.1 Télécharger l'archive "ghost\_in\_the\_shell.tar.gz"

```
wget -g https://github.com/abraham-ukachi/shell/downloads/ghost_in_the_shell.tar.gz
```

- ``wget`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``-g`` - l'alias que
- ``"https://github...shell.tar.gz"`` - l'alias que nous avons créé dans

## 6.2 Désarchiver le fichier

```
tar -zvf "ghost_in_the_shell.tar.gz"
```

- ``tar`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``-zvf`` - l'alias que
- ``"ghost_in_the_shell.tar.gz"`` - l'alias que nous avons créé dans l

# Job 07



JOB: Maintenant, vous allez approfondir les commandes, avec les caractères suivants “> < >> <<|”, votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux:

Toutes les actions ont été réalisées en une seule commande

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

```
echo "Je suis votre fichier text" > une_commande.txt
```

- ``echo`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``"Je suis votre fichier text"`` - l’alias que
- ``>`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``une_commande.txt`` - l’alias que

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

Il n’existe pas un fichier de source **APT** sur mon OS, alors j’ai compté le nombre de lignes présentes dans le fichier de source “.zshrc” à l’aide de la commande suivante:

```
wc -l < ~/.zshrc > nb_lignes.txt
```

- ``wc`` - l’alias que nous avons créé dans le [job 5.2](#)
- ``-l`` - l’alias que



- `<` - l'alias que nous avons créé dans le [job 5.2](#)
- `~/zshrc` - l'alias que
- `>` - l'alias quojokjko kjok
- `nb_lignes.txt` - l'alias que

### 7.3 Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé "save\_sources"

```
cat ~/zshrc | tee save_sources
```

- `cat` - l'alias que nous avons créé dans le [job 5.2](#)
- `~/zshrc` - l'alias que
- `|` - l'alias que nous avons créé dans le [job 5.2](#)
- `tee` - l'alias que
- `save_sources` - l'alias quojokjko kjok

### 7.4 Faites une recherche des fichiers commençant par "." tout en cherchant le mot alias qui sera utilisé depuis un fichier

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

- `ls` - l'alias que nous avons créé dans le [job 5.2](#)
- `-d .*` - l'alias que
- `|` - l'alias que nous avons créé dans le [job 5.2](#)
- `grep` - l'alias que
- `alias` - l'alias quojokjko kjok

# Pour aller plus loin



JOB: Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux en utilisant seulement les caractères suivants “| || & &&”

Toutes les actions ont été réalisées en une seule commande

## 8.1 Installer la commande tree

```
brew install tree
```

- ``brew`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``install`` - l'alias que
- ``tree`` - l'alias que nous avons créé dans le [job 5.2](#)

## 8.2 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”

```
brew / > tree.save &
```

- ``brew`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``/`` - l'alias que
- ``>`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``tree.save`` - l'alias que
- ``&`` - l'alias quojokjko kjok

### 8.3 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 | wc -l
```

- ``ls`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``|`` - l'alias que
- ``wc`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``-l`` - l'alias que

### 8.4 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

Nos aliases du dessus ont été maintenu, alors à l'aide de la commande suivante, nous pouvons lancer une update et si l'update réussit, lancer une upgrade:

```
update && upgrade
```

- ``update`` - l'alias que nous avons créé dans le [job 5.2](#)
- ``&&`` -
- ``upgrade`` - l'alias que nous avons créé dans le [job 5.3](#)

# Questions

## → Comment ajouter des options à une commande ?

Pour afficher des options à une commande il suffit de faire tirtet `-`` qui se nomme flag + l'option représentée par une lettre ``a``. Il n'est pas nécessaire d'ajouter un flag après la première option puisqu'elles se cumulent s'il s'agit d'un paramètre à une seule lettre, comme on le voit avec la commande utilisée dans la capture d'écran du [Job1.2](#).

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

Pour une commande avec un seul paramètre ou argument, il suffit d'un d'entrer *la commande + un flag et le paramètre ou argument*. Lorsqu'on utilise un paramètre avec plus d'une lettre il faudra écrire *la commande + `--`` le paramètre ou argument*.