

~\$hell

Abraham UKACHI

9 Octobre 2022

Sommaire

Introduction	5		
macOS	7	Job 06	24
Job 01	8	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		Job 07	25
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	10	25	
Job 02	11	7.2 Compter le nombre de lignes	
2.1 Lisez un fichier en utilisant		présentes dans votre fichier de	
une commande qui permet		source apt et les enregistrer dans	
seulement de lire	11	un fichier nommé "nb_lignes.txt"	25
2.2 Afficher les 10 premières		7.3 Afficher le contenu du fichier	
lignes du fichier ".bashrc"	12	source apt et l'enregistrer dans un	
2.3 Afficher les 10 dernières lignes		autre fichier appelé	
du fichier ".bashrc"	13	"save_sources"	26
2.4 Afficher les 20 premières lignes		7.4 Faites une recherche des	
du fichier ".bashrc"	13	fichiers commençant par "." tout en	
2.5 Afficher les 20 dernières lignes		cherchant le mot alias qui sera	
du fichier ".bashrc"	14	utilisé depuis un fichier	26
Job 03	15	Pour aller plus loin	26
3.1 Installer le paquet "cmatrix"	15	8.1 Installer la commande tree	26
3.2 lancer le paquet que vous		8.2 Lancer la commande tree en	
venez d'installer	15	arrière-plan qui aura pour but	
3.3 Mettre à jour son gestionnaire		d'afficher toute l'arborescence en	
de paquets	16	de votre / en enregistrant le	
3.4 Mettre à jour ses différents		résultat dans un fichier "tree.save"	
logiciels	16	27	
3.5 Télécharger les internets :		8.3 lister les éléments présents	
Google	17	dans le dossier courant est utilisé	
3.6 Redémarrer votre machine	17	directement le résultat de votre	
3.7 Éteindre votre machine	17	première commande pour compter	
Job 04	18	le nombre d'éléments trouvés	27
4.1 Créer un fichier users.txt qui		8.4 Lancer une commande pour	
contiendra User1 et User2 séparé		update vos paquets, si l'update	
par un retour à la ligne	18	réussi alors, vous devrez lancer un	
4.2 Créer un groupe appelé		upgrade de vos paquets. Si	
"Plateformeurs"	18	l'update échoue, votre upgrade ne	
		se lancera pas	27
		Questions	28

4.3 Créer un utilisateur appelé "User1"	19	Comment ajouter des options à une commande ?	28
4.4 Créer un utilisateur appelé "User2"	19	Quelles sont les deux syntaxes principales d'écriture des options pour une commande ?	28
4.5 Ajouter "User2" au groupe Plateformeurs	20		
4.6 Copier votre "users.txt" dans un fichier "droits.txt"	20		
4.7 Copier votre "users.txt" dans un fichier "groupe.txt"	20		
4.8 Changer le propriétaire du fichier "droits.txt" pour mettre "User1"	20		
4.9 Changer les droits du fichier "droits.txt" pour que "User2" ai accès seulement en lecture	21		
4.10 Changer les droits du fichier "groupes.txt" pour que les utilisateurs puissent accéder au fichier en lecture uniquement	21		
4.11 Changer les droits du fichier pour que le groupe "Plateformeurs" puissent y accéder en lecture/écriture.	22		
Job 05	22		
5.1 Ajouter un alias qui permettra de lancer la commande "ls -la" en tapant "la"	22		
5.2 Ajouter un alias qui permettra de lancer la commande "apt-get update" en tapant "update"	22		
5.3 Ajouter un alias qui permettra de lancer la commande "apt-get upgrade" en tapant "upgrade"	22		
5.4 Ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à votre nom d'utilisateur	23		
5.5 Mettre à jour les modifications de votre bashrc dans votre shell actuel	23		
5.6 Afficher les variables d'environnement	23		
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

Nous pouvons afficher le manuel de la command `ls` avec cette commande:

```
1 | 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@abcdefghiklmnopqrstuvwxyz] [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.

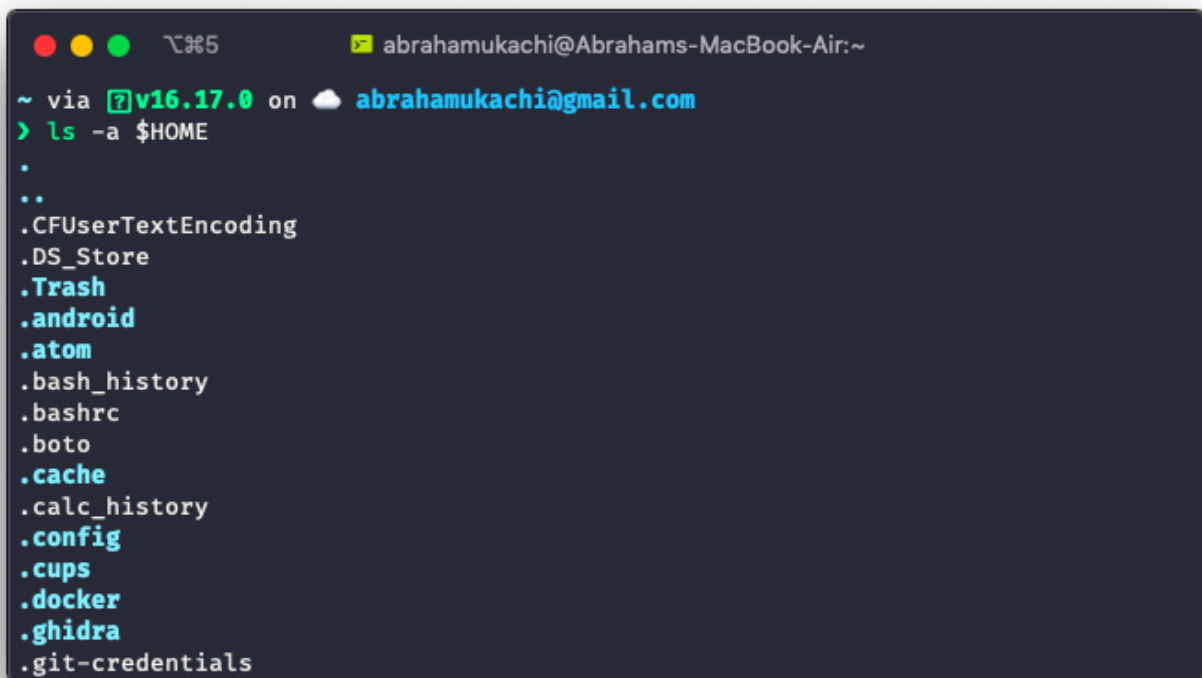
  If no operands are given, the contents of the current directory are dis-
  played. If more than one operand is given, non-directory operands are
  :
  
```

1.2 Afficher les fichiers cachés du home de votre utilisateur

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

```
1 | 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

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

```
1 | 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 :

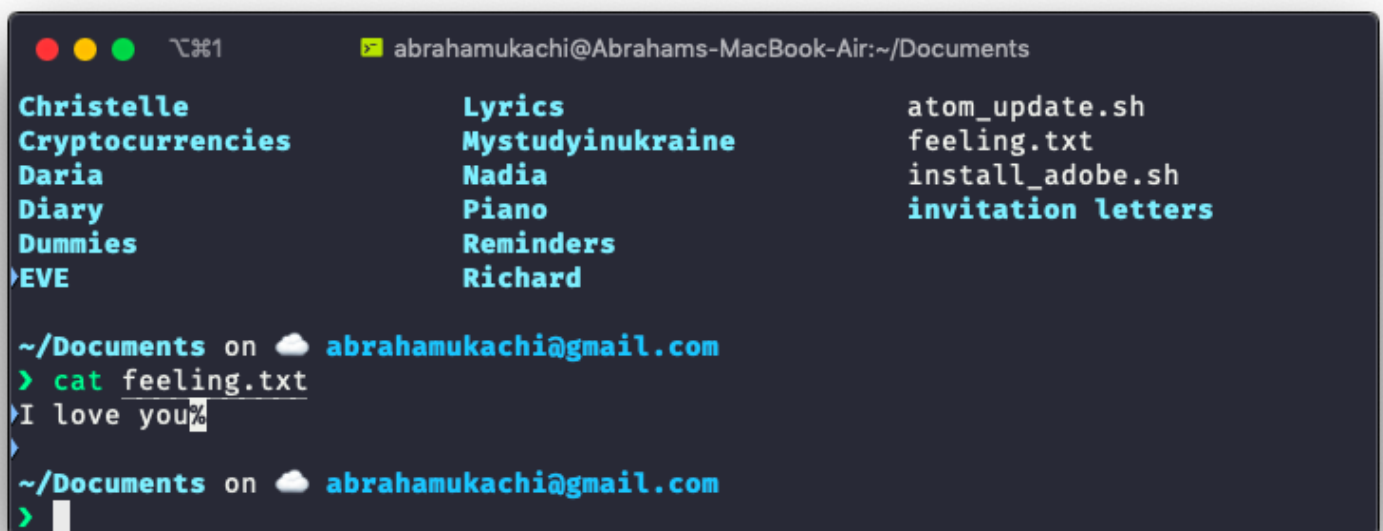
2.1 Lisez un fichier en utilisant une commande qui permet seulement de lire

Nous pouvons lire un fichier sur Linux/UNIX avec la commande suivante:

```
1 | 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:



```
abrahamukachi@Abrahams-MacBook-Air:~/Documents  
Christelle      Lyrics          atom_update.sh  
Cryptocurrencies Mystudyinukraine feeling.txt  
Daria          Nadia          install_adobe.sh  
Diary          Piano          invitation letters  
Dummies        Reminders  
EVE            Richard  
  
~/Documents on cloud abrahamukachi@gmail.com  
> cat feeling.txt  
I love you%  
  
~/Documents on cloud abrahamukachi@gmail.com  
>
```

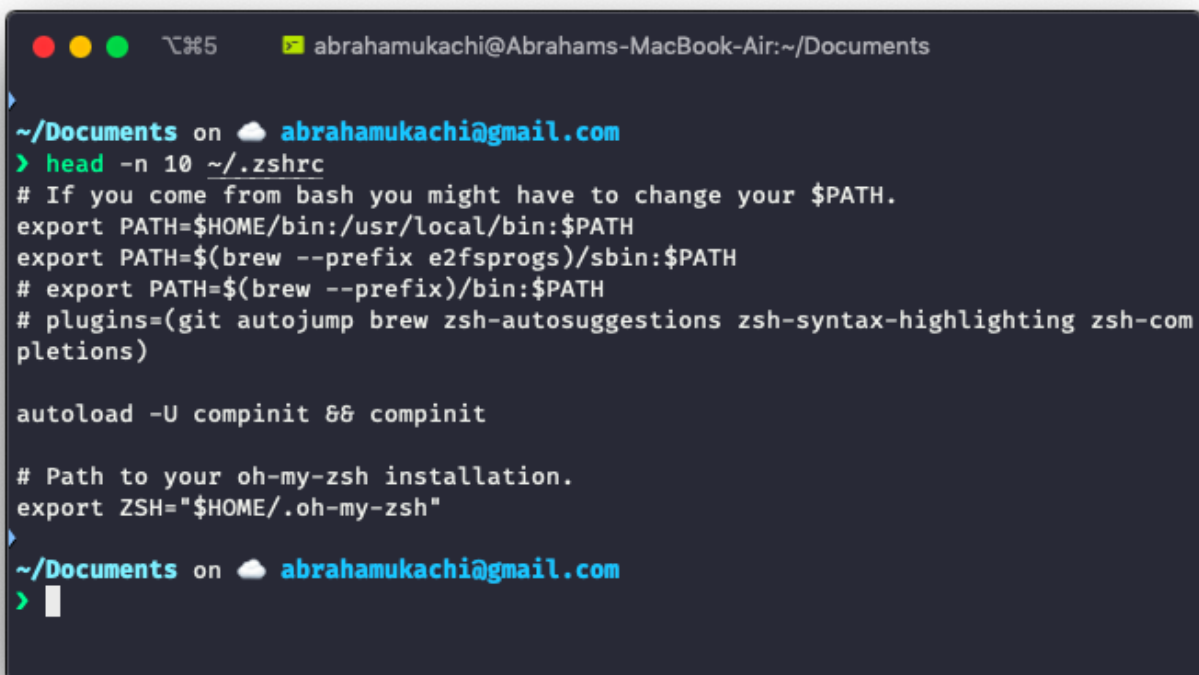
2.2 Afficher les 10 premières lignes du fichier “.bashrc”

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

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

```
1 | 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 “.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.

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

```
1 | 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 “.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.

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

```
1 | 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``

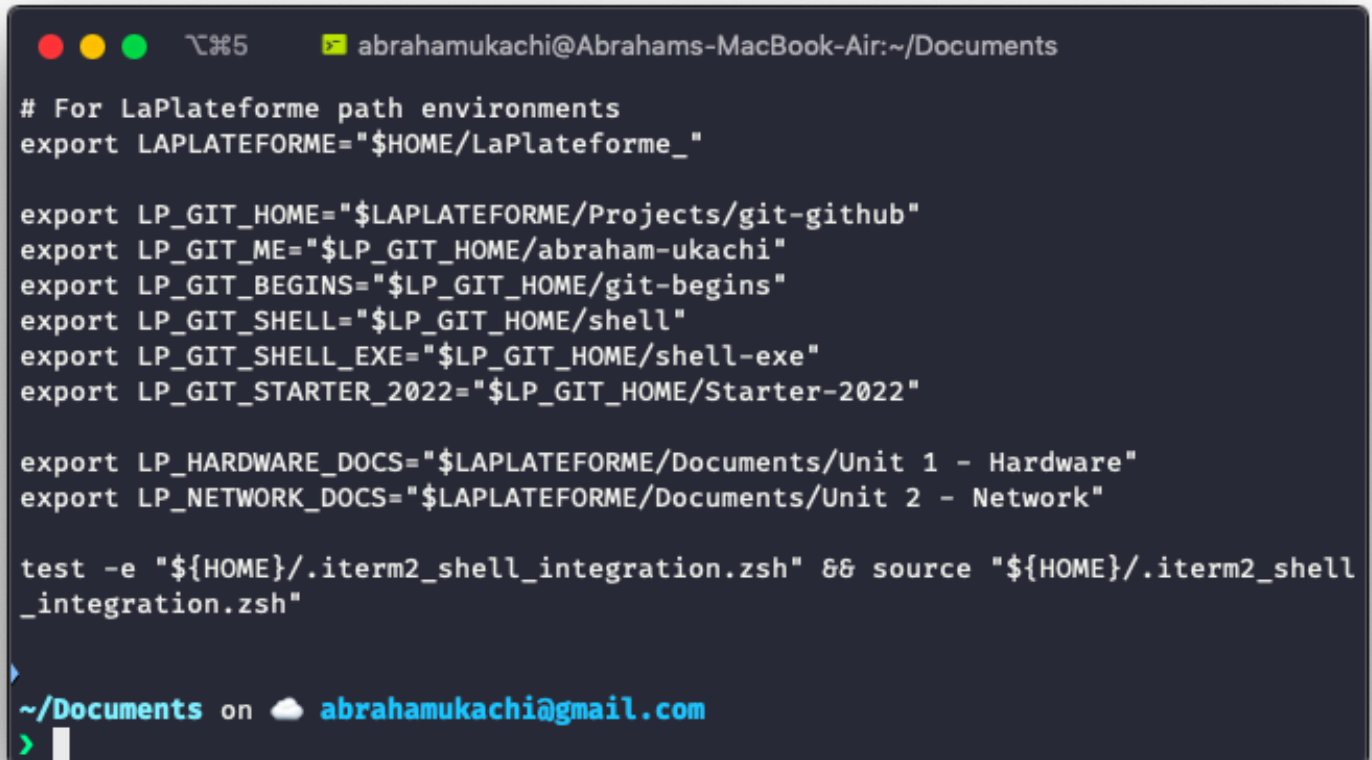
2.5 Afficher les 20 dernières lignes du fichier “.bashrc”

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

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

```
1 | 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`



```


# For LaPlateforme path environments
export LAPLATEFORME="$HOME/LaPlateforme_"

export LP_GIT_HOME="$LAPLATEFORME/Projects/git-github"
export LP_GIT_ME="$LP_GIT_HOME/abraham-ukachi"
export LP_GIT_BEGINS="$LP_GIT_HOME/git-begins"
export LP_GIT_SHELL="$LP_GIT_HOME/shell"
export LP_GIT_SHELL_EXE="$LP_GIT_HOME/shell-exe"
export LP_GIT_STARTER_2022="$LP_GIT_HOME/Starter-2022"

export LP_HARDWARE_DOCS="$LAPLATEFORME/Documents/Unit 1 - Hardware"
export LP_NETWORK_DOCS="$LAPLATEFORME/Documents/Unit 2 - Network"

test -e "${HOME}/.iterm2_shell_integration.zsh" && source "${HOME}/.iterm2_shell_integration.zsh"

```

~/Documents on  abrahamukachi@gmail.com

Une partie des 20 dernière lignes du fichier “.zshrc”

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”

Nous pouvons installer le paquet **cmatrix** à l’aide de la commande suivante:

```
1 | brew install cmatrix
```

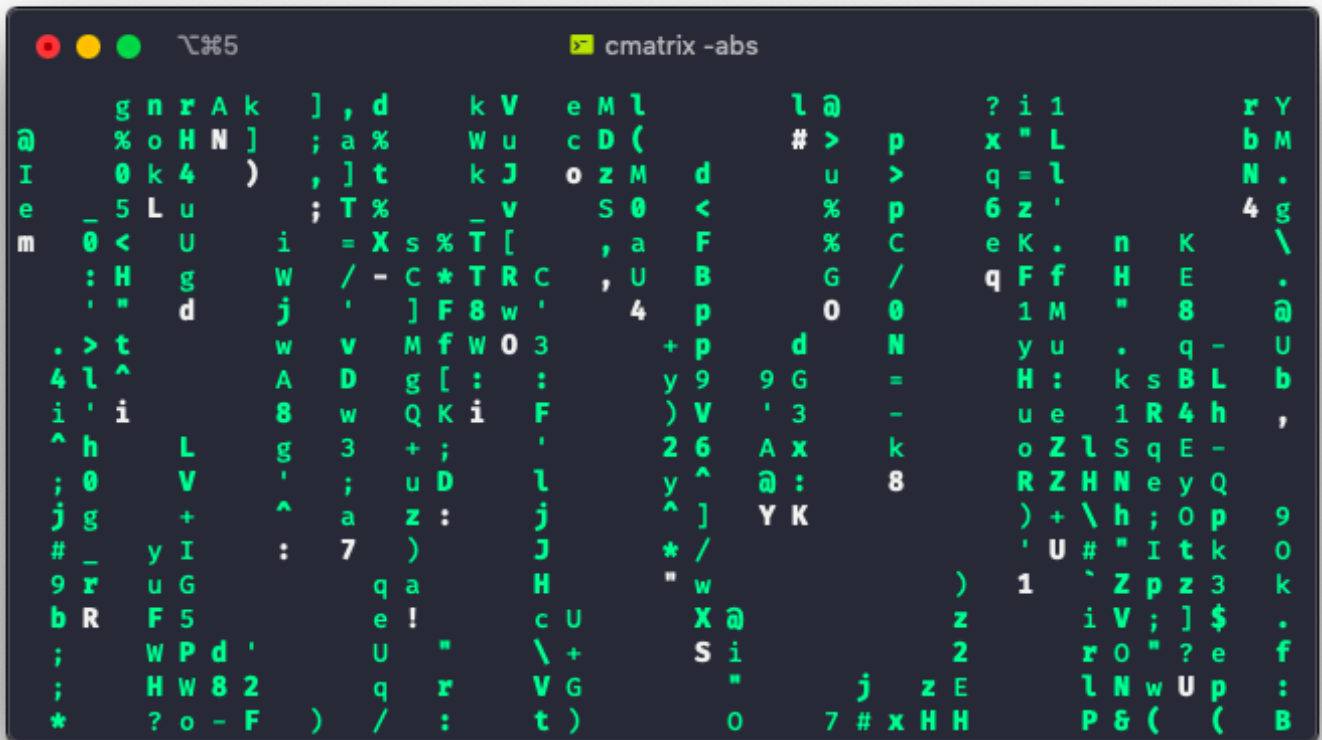
- **`brew`** - (*commande*) gestionnaire de paquets gratuit et open-source qui s’installe sur macOS qui permet d’installer en ligne de commande des logiciels/applications.
- **`install`** - (*commande de brew*) install des paquets ou applications.
- **`cmatrix`** - (*paquet*) commande qui sera installée par **`brew`**.

3.2 lancer le paquet que vous venez d’installer

Nous pouvons lancer le paquet que nous venons d’installer avec cette commande:

```
1 | cmatrix
```

- **`cmatrix`** - (*commande*) affiche une matrice de style néo sur le terminal.



< **cmatrix** > avec les options **`abs`**

3.3 Mettre à jour son gestionnaire de paquets

Nous pouvons mettre à jour le gestionnaire de paquets **Homebrew** ou **brew** à l'aide de la command suivante:

```
1 | brew update
```

- **`brew`** - (*commande*) gestionnaire de paquets sur macOS.
- **`update`** - (*commande de brew*) pour mettre **`brew`** à jour.

3.4 Mettre à jour ses différents logiciels

Nous pouvons lancer le paquet que vous venez d'installer avec cette commande:

```
brew upgrade
```

- ``brew`` - (*commande*) gestionnaire de paquets sur macOS.
- ``upgrade`` - (*commande de brew*) pour mettre à jour ses logiciels/casks.

3.5 Télécharger les internets : Google

Nous pouvons télécharger les internets Google à l'aide de cette commande:

```
1 | wget -r www.google.com
```

- ``wget`` - (*commande*) pour télécharger des fichiers depuis l'Internet en utilisant les protocoles HTTP, HTTPS et FTP.
- ``-r`` - (*option*) pour récupérer les fichiers récursivement.
- ``www.google.com`` - (*URL*) le lien ou site à télécharger.

3.6 Redémarrer votre machine

Nous pouvons redémarrer notre machine à l'aide de cette commande:

```
1 | sudo reboot
```

- ``sudo`` - (*commande*) permet à tout utilisateur d'exécuter une commande ou de restreindre tout utilisateur.
- ``reboot`` - (*commande*) pour redémarrer la machine.

3.7 Éteindre votre machine

Nous pouvons éteindre notre machine à l'aide de cette commande:

```
1 | sudo shutdown -h now
```

- ``sudo`` - (*commande*) permet à tout utilisateur d'exécuter une commande ou de restreindre tout utilisateur.
- ``shutdown`` - (*commande*) pour éteindre la machine.

- `-h now` - Le `-h` (*option*) renvoie explicitement au fait d'éteindre ou de désactiver le système, et avec `now` (*indication temporelle*), la machine s'arrête immédiatement.

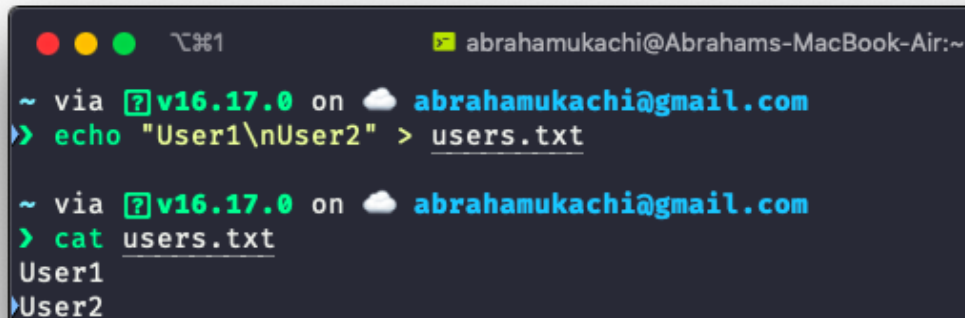
Job 04



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

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



```

~ via [?] v16.17.0 on [cloud] abrahamukachi@gmail.com
> echo "User1\nUser2" > users.txt

~ via [?] v16.17.0 on [cloud] abrahamukachi@gmail.com
> cat users.txt
User1
User2

```

4.2 Créer un groupe appelé "Plateformeurs"

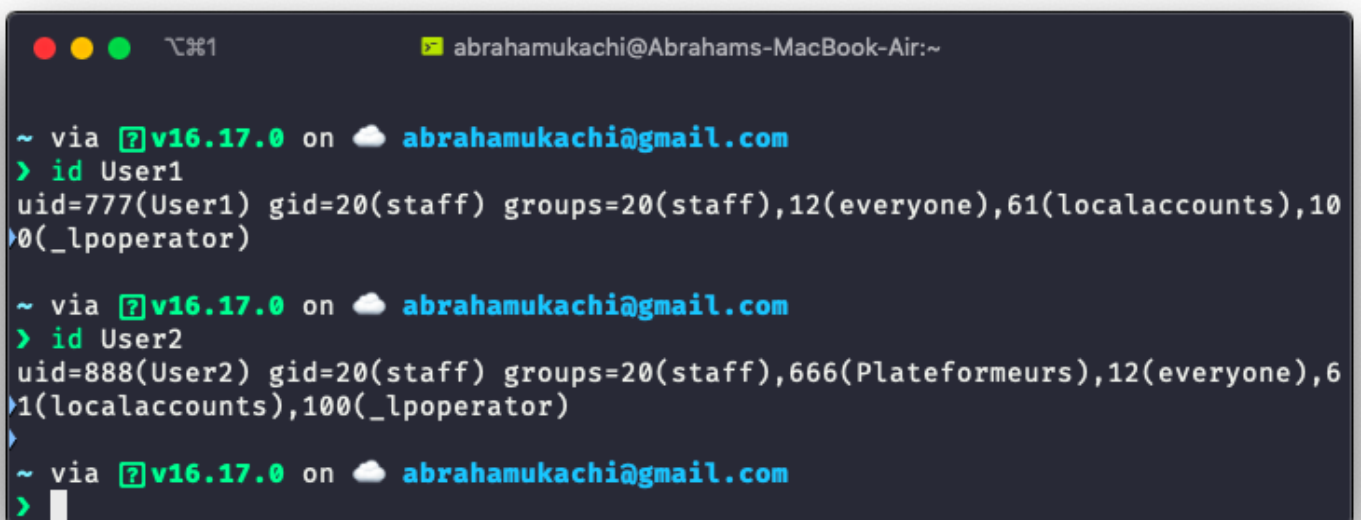
```
1 | sudo dscl . -create /Groups/Plateformeurs
PrimaryGroupID 666
```


4.3 Créer un utilisateur appelé “User1”

```
1 | sudo dsc1 . -create /Users/User1 UniqueID 777
2 | sudo dsc1 . -create /Users/User1 PrimaryGroupID 20
```

4.4 Créer un utilisateur appelé “User2”

```
1 | sudo dsc1 . -create /Users/User2 UniqueID 888
2 | sudo dsc1 . -create /Users/User2 PrimaryGroupID 20
```

A terminal window on a MacBook Air showing the execution of commands to create two users, User1 and User2, and then verifying their creation using the 'id' command. The terminal output shows the user details for User1 (uid=777, gid=20) and User2 (uid=888, gid=20).

```
~ via [?] v16.17.0 on [?] abrahamukachi@gmail.com
> id User1
uid=777(User1) gid=20(staff) groups=20(staff),12(everyone),61(localaccounts),100(_lpoperator)

~ via [?] v16.17.0 on [?] abrahamukachi@gmail.com
> id User2
uid=888(User2) gid=20(staff) groups=20(staff),666(Plateformeurs),12(everyone),61(localaccounts),100(_lpoperator)

~ via [?] v16.17.0 on [?] abrahamukachi@gmail.com
> 
```

4.5 Ajouter “User2” au groupe Plateformeurs

```
1 | sudo dsc1 . -append /Groups/Plateformeurs  
GroupMembership User2
```

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

```
1 | cp users.txt droits.txt
```

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

```
1 | cp users.txt groupes.txt
```

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

```
1 | sudo chown User1 droits.txt
```

```
~ via [?] v16.17.0 on [cloud] abrahamukachi@gmail.com
> cat droits.txt
User1
> User2

~ via [?] v16.17.0 on [cloud] abrahamukachi@gmail.com
> stat -l droits.txt
-> -rw-r--r--  1 User1 Plateformeurs 12 Sep 26 06:52:39 2022 droits.txt
>

~ via [?] v16.17.0 on [cloud] abrahamukachi@gmail.com
> 
```

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
```

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

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

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
```

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”

```
1 | alias la="ls -la"
```

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

```
1 | alias update="brew update"
```

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

```
1 | alias upgrade="brew upgrade"
```

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

```
1 | export USER="abrahamukachi"
```

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

```
1 | source ~/.zshrc
```

5.6 Afficher les variables d'environnement

```
1 | printenv
```

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

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

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"

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

6.2 Désarchiver le fichier

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

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”

```
1 | echo "Je suis votre fichier texte" > une_commande.txt
```

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:

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

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

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

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

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

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

```
1 | brew install tree
```


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"

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

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

```
1 | ls | wc -l
```

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:

```
1 | update && upgrade
```

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*.