

SHELL

JOB 1

- Afficher le manuel de la commande ls

```
lucas@debian:~$ ls
Bureau  Documents  Images  Modèles  Musique  Public  Téléchargements  Vidéos
lucas@debian:~$
```

- Afficher les fichiers cachés du home de votre utilisateur

```
lucas@debian:~$ ls
Bureau  Documents  Images  Modèles  Musique  Public  Téléchargements  Vidéos
lucas@debian:~$ ls -a
.          .bash_logout  .cache  Images  Musique  Téléchargements
..         .bashrc       .config .local  .profile  Vidéos
.bash_history  Bureau      Documents  Modèles  Public
lucas@debian:~$
```

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

```
lucas@debian:~$ ls -la
total 68
drwxr-xr-x 13 lucas lucas 4096 29 sept. 10:48 .
drwxr-xr-x  3 root  root  4096 29 sept. 10:07 ..
-rw-r--r--  1 lucas lucas   47 29 sept. 10:48 .bash_history
-rw-r--r--  1 lucas lucas  220 29 sept. 10:07 .bash_logout
-rw-r--r--  1 lucas lucas 3526 29 sept. 10:07 .bashrc
drwxr-xr-x  2 lucas lucas 4096 29 sept. 10:09 Bureau
drwx-----  9 lucas lucas 4096 29 sept. 10:27 .cache
drwx----- 10 lucas lucas 4096 29 sept. 10:18 .config
drwxr-xr-x  2 lucas lucas 4096 29 sept. 10:09 Documents
drwxr-xr-x  2 lucas lucas 4096 29 sept. 10:09 Images
drwxr-xr-x  3 lucas lucas 4096 29 sept. 10:09 .local
drwxr-xr-x  2 lucas lucas 4096 29 sept. 10:09 Modèles
drwxr-xr-x  2 lucas lucas 4096 29 sept. 10:09 Musique
-rw-r--r--  1 lucas lucas   807 29 sept. 10:07 .profile
drwxr-xr-x  2 lucas lucas 4096 29 sept. 10:09 Public
drwxr-xr-x  2 lucas lucas 4096 29 sept. 10:09 Téléchargements
drwxr-xr-x  2 lucas lucas 4096 29 sept. 10:09 Vidéos
lucas@debian:~$
```

Comment ajouter des options à une commande ?

Pour ajouter des options à une commande, on utilise “-”.

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

les deux syntaxes sont “-” pour faire une abréviation exemple “-a”
ou “--” qui est juste le nom complet “--all”

JOB 2

pour afficher les 10 premières lignes d’un fichier on utilise la commande head suivi du nom du fichier qu’on veut lire :

head .bashrc

```
lucas@debian:~$ 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

lucas@debian:~$
```

Pour les 10 dernières lignes on remplace la commande head par la commande tail.

```
lucas@debian:~$ 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

lucas@debian:~$
```

pour afficher un nombre de lignes spécifique on utilise la commande head -n suivi du nombre de ligne puis du nom du fichier :

head -n 20 .bashrc

```
lucas@debian:~$ 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
lucas@debian:~$
```

Même chose avec la commande tail pour les 20 dernières lignes

```
lucas@debian:~$ 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
lucas@debian:~$
```

JOB 3

La commande pour installer le paquet cmatrix est :

```
sudo apt-get install cmatrix
```

La commande pour mettre à jour le paquet est :

```
sudo apt-get upgrade cmatrix
```

La commande pour mettre à jour ses différents logiciels est :

```
sudo apt-get update
```

Pour éteindre son pc on utilise la commande `sudo shutdown -h now`

Pour redémarrer son pc on utilise la commande `sudo reboot now`

JOB 4

- Créer un groupe appelé "Plateformeurs"

- Créer un utilisateur appelé "User1"

- Créer un utilisateur appelé "User2"

```
touch users.txt
```

```
nano users.txt
```

```
ecrire : User1
```

```
        User2
```

```
sudo groupadd Plateformeurs
```

```
sudo useradd User1
```

```
sudo useradd User2
```

- Ajouter "User2" au groupe Plateformeurs

```
sudo usermod -a -G Plateformeurs User2
```

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

touch droits.txt

cp -r users.txt droits.txt

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

cp -r users.txt groupes.txt

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

sudo chown -c User1 droits.

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

sudo chmod g=r droits.txt

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

sudo chmod ug=r groupes.txt

- Changer les droits du fichier pour que le groupe "Plateformeurs" puisse y accéder en lecture/écriture.

sudo chown :Plateformeurs groupes.txt

sudo chmod g+w groupes.txt

JOB 5

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

nano .bash_aliases

alias la='ls -la'

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

```
nano .bash_aliases
```

puis rajouter a la ligne :

```
alias update='sudo apt-get update'
```

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

```
nano .bash_aliases
```

puis rajouter a la ligne :

```
alias upgrade='sudo apt-get upgrade'
```

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

```
nano .bashrc
```

tout en bas je rajoute :

```
export USER=luкас
```

- Mettre à jour les modifications de votre bashrc dans votre shell actuel
exec bash

- Afficher les variables d'environnement

```
env
```

JOB 6

Pour le désarchiver il faut d'abord le télécharger sur chrome puis taper la commande :

```
tar -vfx 'Ghost in the Shell.tar.gz'
```

JOB 7

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

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

- 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
```

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

```
cp /etc/apt/sources.list $USER /save_sources.txt
```

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

```
grep --include=* alias --include=* . -rnw
```

commande complète :

```
echo "Je suis votre fichier texte"> une_commande.txt | wc -l  
/etc/apt/sources.list > nb_lignes.txt | cat /etc/apt/sources.list | cp  
/etc/apt/sources.list $USER /save_sources.txt | grep --include=* alias  
--include=* . -rnw
```

Pour aller plus loin

```
sudo apt-get install tree & tree / > tree.save | update && upgrade
```

Exercice Bonus (pas fini)

Avant d'installer SSH sur votre machine, assurez-vous qu'elle soit à jour:

```
sudo apt update && sudo apt upgrade -y
```

Ensuite, il faut installer le paquet openssh-server :

```
sudo apt install openssh-server
```

Puis on active et on démarre le service SSH :

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

puis on génère une clé

```
ssh-keygen -t rsa
```

```
lucas@debian:~$ ssh lucas@192.168.229.130
The authenticity of host '192.168.229.130 (192.168.229.130)' can't be established.
ECDSA key fingerprint is SHA256:urCG0EtFJTRNdSrD0RyfxLpMH7M9Eu4tXsar/a1vrYU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/home/lucas/.ssh/known_hosts)
.
lucas@192.168.229.130's password:
Linux debian 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
lucas@debian:~$
```