

JOB 1

Pour afficher le manuel de la commande ls, j'ai utilisé la commande "man ls". Ensuite pour afficher les fichiers cachés j'ai utilisé la commande "ls -a" qui va montrer tous les fichiers y compris ceux commençant par un ".". Pour les informations sur les droits on peut utiliser la commande "-l", j'ai donc tapé "ls -al" pour avoir les deux commandes demandées d'un coup.

- Comment ajouter des options à une commande ?

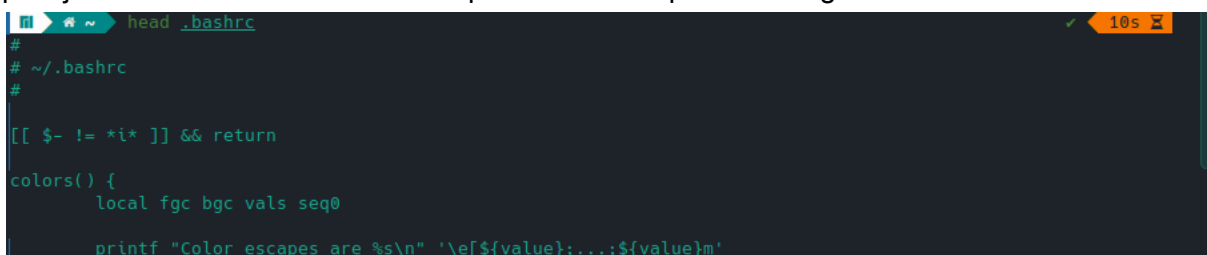
Pour ajouter des options on va utiliser la commande "set -O option"

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

On peut soit utiliser la commande que j'ai écrite ci-dessus "set -O option" ou utiliser son abréviation : "set -abréviation".

JOB 2

Pour lire un fichier on utilise la commande "less". J'ai commencé par lire le fichier ".bashrc" puis j'ai utilisé la commande "head" pour lire les 10 premières lignes du fichier.

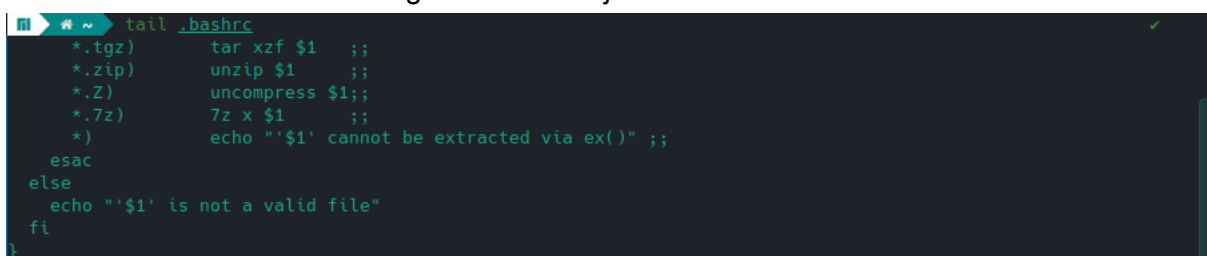


```
head .bashrc
#
# ~/.bashrc
#

[[ $- != *i* ]] && return

colors() {
    local fgc bgc vals seq0
    printf "Color escapes are %s\n" '\e[${value};;...;${value}m'
```

Pour afficher les 10 dernières lignes du fichier j'ai utilisé la commande "tail".



```
tail .bashrc
*.tgz)    tar xzf $1 ;;
*.zip)    unzip $1 ;;
*.Z)      uncompress $1;;
*.7z)     7z x $1 ;;
*)        echo "'$1' cannot be extracted via ex()" ;;
esac
else
    echo "'$1' is not a valid file"
fi
}
```

Pour afficher les 20 premières lignes j'ai utilisé l'option "-n 20".

```
head -n 20 .bashrc
#
# ~/.bashrc
#

[[ $- != *i* ]] && return

colors() {
    local fgc bgc vals seq0

    printf "Color escapes are %s\n" '\e[${value};;...;${value}m'
    printf "Values 30..37 are \e[33mforeground colors\e[m\n"
    printf "Values 40..47 are \e[43mbackground colors\e[m\n"
    printf "Value 1 gives a \e[1mbold-faced look\e[m\n\n"

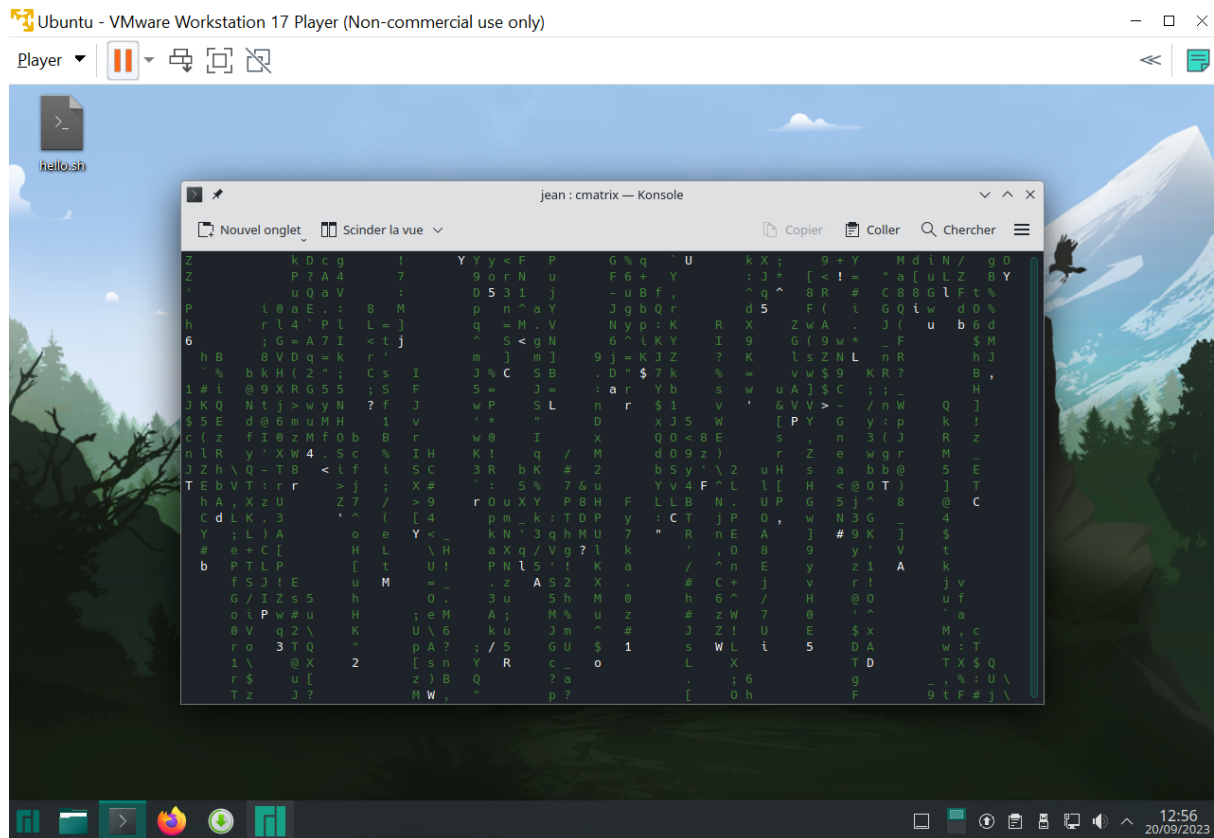
    # foreground colors
    for fgc in {30..37}; do
        # background colors
        for bgc in {40..47}; do
            fgc=${fgc#37} # white
            bgc=${bgc#40} # black
```

Pour les 20 dernières lignes j'ai fait pareil.

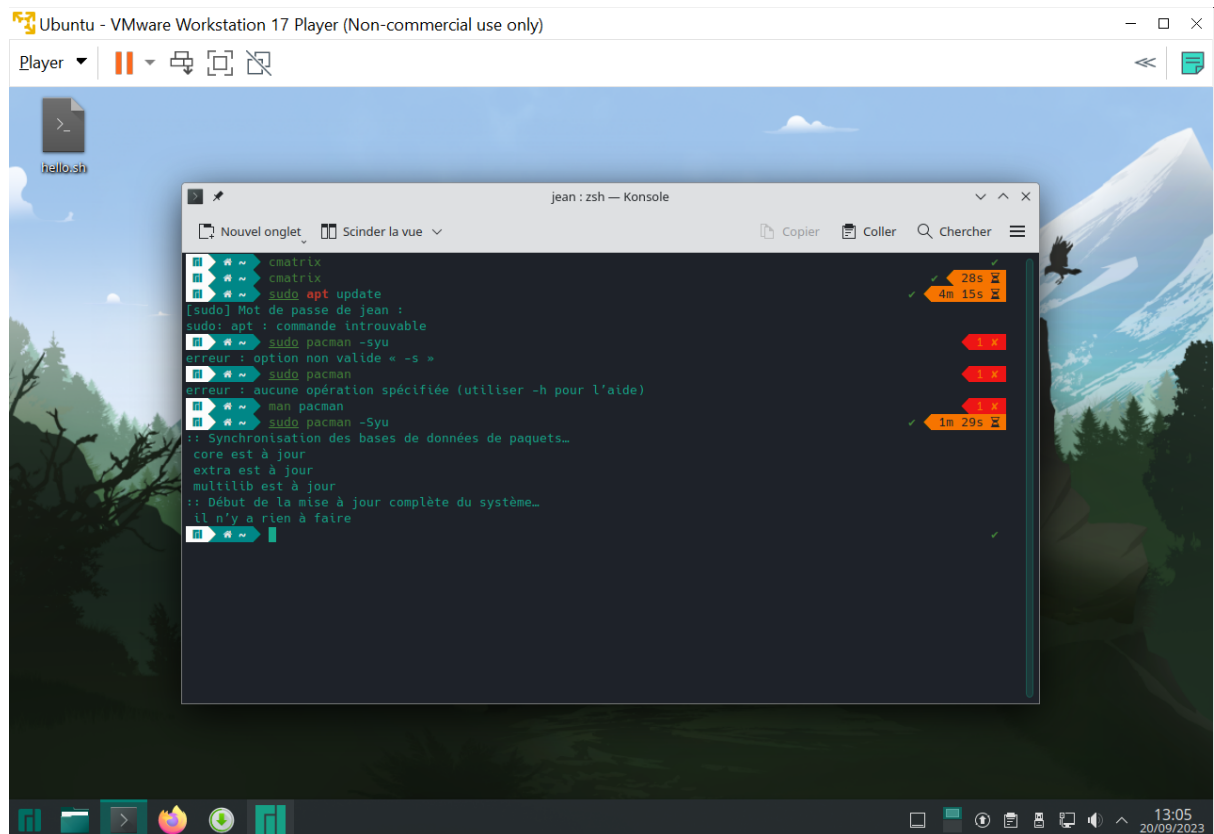
```
tail -n 20 .bashrc
{
    if [ -f $1 ] ; then
        case $1 in
            *.tar.bz2)  tar xjf $1 ;;
            *.tar.gz)   tar xzf $1 ;;
            *.bz2)      bunzip2 $1 ;;
            *.rar)       unrar x $1 ;;
            *.gz)        gunzip $1 ;;
            *.tar)       tar xf $1 ;;
            *.tbz2)      tar xjf $1 ;;
            *.tgz)       tar xzf $1 ;;
            *.zip)       unzip $1 ;;
            *.Z)         uncompress $1;;
            *.7z)        7z x $1 ;;
            *)           echo "'$1' cannot be extracted via ex()" ;;
        esac
    else
        echo "'$1' is not a valid file"
    fi
}
```

JOB 3

Etant sur manjaro, j'ai utilisé la commande "pamac install matrix" pour télécharger et installer le paquet. Je l'ai ensuite lancé en tapant "cmatrix" après avoir redémarré.



Pour mettre à jour mon gestionnaire de paquets j'ai utilisé la commande "sudo pacman -Syu". Cette commande permet aussi de mettre à jour les logiciels. Dans mon cas tout était à jour.



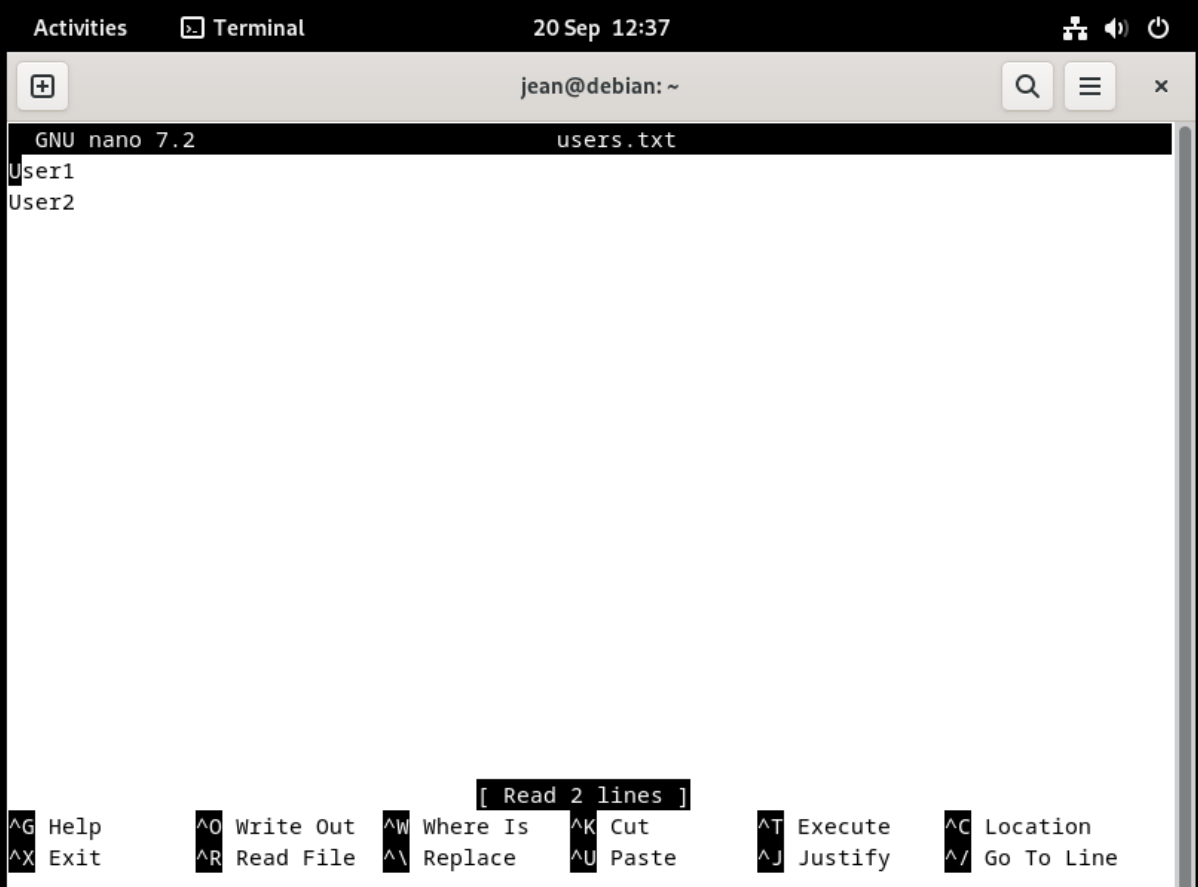
Pour la suite je suis passé de manjaro à debian, je trouvais que manjaro avait trop de particularités et avait l'air plus compliqué, il fallait par exemple pour installer chrome passer par git ou AUR et je ne comprenais pas tout.

En étant sur debian j'ai mis à jour mes paquets comme sur manjaro en utilisant la commande "apt update", j'ai vu que les paquets n'avaient pas besoin d'être mis à jour donc je n'ai pas eu à taper "apt upgrade".

J'ai ensuite téléchargé chrome avec la commande "wget" puis installé avec la commande "sudo apt install".

JOB 4

Pour créer un fichier texte j'ai ensuite utilisé la commande "touch". Pour éditer mon fichier j'ai utilisé la commande "nano". J'ai donc tapé "touch users.txt" puis "nano users.txt". J'ai ensuite créé les deux users séparés d'un retour à la ligne.



```
Activities  Terminal  20 Sep 12:37
jean@debian: ~
GNU nano 7.2  users.txt
User1
User2

[ Read 2 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

J'ai ensuite créé un groupe "Plateformeurs" avec la commande "groupadd Plateformeurs". J'ai ensuite créé les utilisateurs 1 et 2 en faisant la commande "useradd" suivi de "User1" et "User2".

Avec la commande suivante "adduser User2 Plateformeurs" on ajoute l'utilisateur 2 au groupe Plateformeurs.

Pour copier le contenu de "users.txt" vers "droits.txt" et "groupes.txt" on va écrire "cp users.txt droits.txt" et "cp users.txt groupes.txt".

Pour faire de User1 le propriétaire de "droits.txt" on utilise la commande "chown User1 users.txt".

Avec la commande chmod on va pouvoir changer les permissions. Si on veut donner la permission à User2 d'avoir uniquement la lecture du document "users.txt" on peut passer par le groupe Plateformeurs car il en est l'unique membre. On va donc d'abord attribuer le fichier au groupe en tapant "chgrp Plateformeurs users.txt" puis utiliser la commande "chmod g+r users.txt" pour donner au groupe le droit de lecture sans le droit d'écriture (je ne me souviens plus si le groupe avait de base le droit d'écrire dessus dans ce cas là j'ai du faire g-w users.txt").

Pour que les utilisateurs du fichier "groupes.txt" puissent accéder au fichier en lecture uniquement on écrit "chmod a+r groupes.txt" et "chmod u-w groupes.txt" ce qui donnera la bonne permission aux utilisateurs tout en enlevant le droit d'écriture au propriétaire du fichier.

Pour que le groupe plateformeurs puisse y accéder en lecture/écriture on écrit "chgrp Plateformeurs groupes.txt" puis "chmod g+w groupes.txt".

Voilà à quoi ressemblent les droits et permissions de mes fichiers à la fin :

```
root@debian:~# ls -l
total 101052
-rwxr--r-- 1 User1 Plateformeurs      12 Sep 20 12:57 droits.txt
-rw-r--r-- 1 root  root                103464324 Sep 15 03:33 google-chrome-stable_current_amd64.deb
-r--rw-r-- 1 root  Plateformeurs      12 Sep 20 12:58 groupes.txt
-rw-r--r-- 1 root  root                12 Sep 20 12:57 users.txt
root@debian:~# █
```

JOB 5

Pour créer des alias je les ai rentrés dans les fichiers suivants : "~/.bash_aliases" et "~/.bashrc". J'ai ensuite rajouté la commande suivante qui manquait dans mon .bashrc :

```
if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi
```

J'ai ensuite utilisé la commande "source" sur les deux fichiers puis redémarré. Mes alias fonctionnent au redémarrage.

J'ai ensuite créé la variable d'environnement suivante : "USER=jean".

Pour la recharger j'ai utilisé la commande "source .bashrc".

Pour afficher les variables d'environnement j'ai utilisé la commande "printenv" et cela m'a affiché les variables suivantes :

```
root@debian:~# printenv
SHELL=/bin/bash
LANGUAGE=en_GB:en
PWD=/root
LOGNAME=root
HOME=/root
LANG=en_GB.UTF-8
TERM=xterm-256color
USER=jean
SHLVL=1
XDG_DATA_DIRS=/usr/share/gnome:/usr/local/share/:/usr/share/
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MAIL=/var/mail/root
OLDPWD=/
_=/usr/bin/printenv
root@debian:~# source .bashrc
root@debian:~# exec bash
root@debian:~# █
```

Ensuite pour modifier le PATH j'ai édité mon fichier .bashrc en mode root-user avec la commande nano. J'ai ajouté la ligne de code suivante : export PATH="\$HOME/jean/Desktop:\$PATH".

JOB 6

J'ai ensuite téléchargé le fichier 'Ghost in the Shell'.tar depuis l'interface graphique Debian. Depuis le terminal je suis ensuite allé dans le fichier "Downloads" où je l'ai téléchargé et j'ai tapé la commande "tar -xvf 'Ghost in the Shell.tar'" ce qui m'a décompressé le fichier en pdf.

JOB 7

Pour créer un fichier "une_commande.txt" avec le texte suivant "Je suis votre fichier texte", compter le nombre de lignes présentes dans mon fichier de source apt et les enregistrer dans un fichier nommé "nb_lignes.txt", afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé "save_sources" et faire une recherche des fichiers commençants par "." tout en cherchant le mot alias qui sera utilisé depuis un fichier, j'ai utilisé la commande suivante, avec le résultat :

```
jean@debian:~$ echo "Je suis votre fichier texte" > une_commande.txt | wc -l | /etc/apt/sources.list > nb_lignes.txt | less /etc/apt/sources.list > save_sources.txt | find ./* | grep -r alias
grep: ./mozilla/firefox/zg9v6k6j.default-esr/storage/permanent/chrome/ldb/3870112724rsegmolttet-es.sqlite: binary file matches
grep: ./mozilla/firefox/zg9v6k6j.default-esr/storage/default/https+mail.google.com/cache/mozgue/177/0215370b-f385-4468-a753-623f93566bb1.final: binary file matches
grep: ./mozilla/firefox/zg9v6k6j.default-esr/storage/default/https+mail.google.com/cache/mozgue/49/482fced8-4ebe-4e27-a172-d1d5a464de45.final: binary file matches
grep: ./mozilla/firefox/zg9v6k6j.default-esr/storage/default/https+mail.google.com/cache/mozgue/239/6537725f-344d-43e3-9dc2-205e6d76e5ef.final: binary file matches
grep: ./mozilla/firefox/zg9v6k6j.default-esr/storage/default/https+mail.google.com/cache/mozgue/177/34ed482a-a4dd-468d-bed9-d2df09ae8e11.final: binary file matches
grep: ./mozilla/firefox/zg9v6k6j.default-esr/storage/default/https+mail.google.com/cache/mozgue/195/86a87876-b2fc-447f-a519-3c8135583ac3.final: binary file matches
bash_history:find ./ | grep alias
bash_history:find -name .* | grep alias
bash_history:sudo find -name .* | grep alias
bash_history:sudo find ./ | grep -r 'alias'
bash_history:sudo find ./ | gre -r 'alias'
bash_history:sudo find ./ | grep -r 'alias'
bash_history:sudo find ./ | grep -r alias
bash_history:sudo find -name .* | grep -r alias
bash_history:sudo find ./ | grep -r alias
bash_history:sudo find ./ | grep -r 'alias'
bash_history:sudo find .* | grep -r alias
bash_history:sudo find -.* | grep -r alias
bash_history:sudo find -.* | grep alias
bash_history:sudo find ./ | grep alias
bash_history:sudo find ./ | grep alias
bash_history:find -name .* | grep -r alias
bash_history:find .* | grep -r alias
bash_history:find -name .* | grep -r 'alias'
bash_history:find -name .* | grep -r 'alias'
bash_history:find -name .* | grep -r 'alias'
bash_history:find -name .* | grep 'alias'
bash_history:find -name .* | grep 'alias'
bash_history:find -name .* | grep alias
grep: ./cache/mozilla/firefox/zg9v6k6j.default-esr/cache2/entries/EBA3112825C17C88E080830808E4928C25C1F0: binary file matches
grep: ./cache/mozilla/firefox/zg9v6k6j.default-esr/startupCache/scriptCache.bin: binary file matches
grep: ./cache/mozilla/firefox/zg9v6k6j.default-esr/startupCache/webext.sc.124: binary file matches
grep: ./cache/mozilla/firefox/zg9v6k6j.default-esr/startupCache/scriptCache-current.bin: binary file matches
grep: ./cache/riscv/riscv/files/https://www.kernel.org/pub/linux/kernel/v5.x/doc/riscv64/riscv64-64bit-abi.pdf: binary file matches
grep: ./cache/gnome-software/odrs/ratings.json: "is_snapcraft_allianmon-25M54ow32KRCUW2Y3Hd5BuaBF1fg8": {
grep: ./cache/gnome-software/appstream/components.xml.b: binary file matches
grep: ./cache/gstreamer-1.0/registry-08_04.bin: binary file matches
bashrc:# enable color support of ls and also add handy aliases
bashrc: alias ls='ls --color=auto'
bashrc: alias dir='dir --color=auto'
bashrc: alias vdir='vdir --color=auto'
bashrc: alias grep='grep --color=auto'
bashrc: alias fgrep='fgrep --color=auto'
bashrc: alias egrep='egrep --color=auto'
bashrc:# some more ls aliases
bashrc:alias ll='ls -l'
bashrc:alias la='ls -A'
bashrc:alias l='ls -CF'
bashrc:# -/./, bash_aliases, instead of adding them here directly.
bashrc:if [ -f ~/.bash_aliases ]; then
bashrc: . ~/.bash_aliases
```

POUR ALLER PLUS LOIN

Pour installer la commande tree, lancer la commande tree en arrière-plan, enregistrer le résultat dans un fichier “tree.save”, lister les éléments présents dans le dossier courant et utiliser directement le résultat de ma première commande pour compter le nombre d’éléments trouvés puis lancer une commande pour update mes paquets, si l’update réussit lancer un upgrade de mes paquets, j’utilise la commande suivante :

```
root@debian:~# apt install tree | tree ~/ & > tree.save | ls > wc -l | apt update && apt upgrade
[1] 5700
/root/
├── droits.txt
├── google-chrome-stable_current_amd64.deb
├── groupes.txt
├── tree.save
├── users.txt
└── wc

1 directory, 6 files
```

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

```
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Hit:4 https://dl.google.com/linux/chrome/deb stable InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
[1]+  Done                  apt install tree | tree ~/
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@debian:~#
```