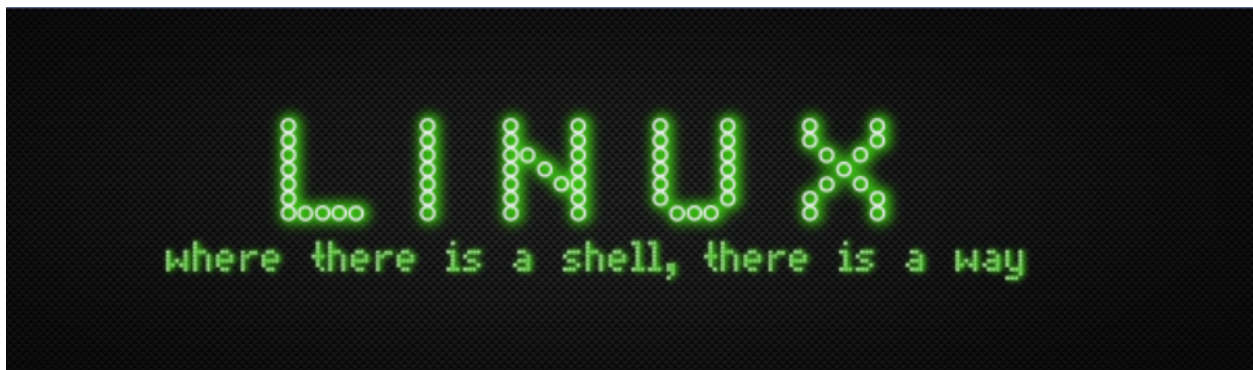


Documentation Shell

Shell

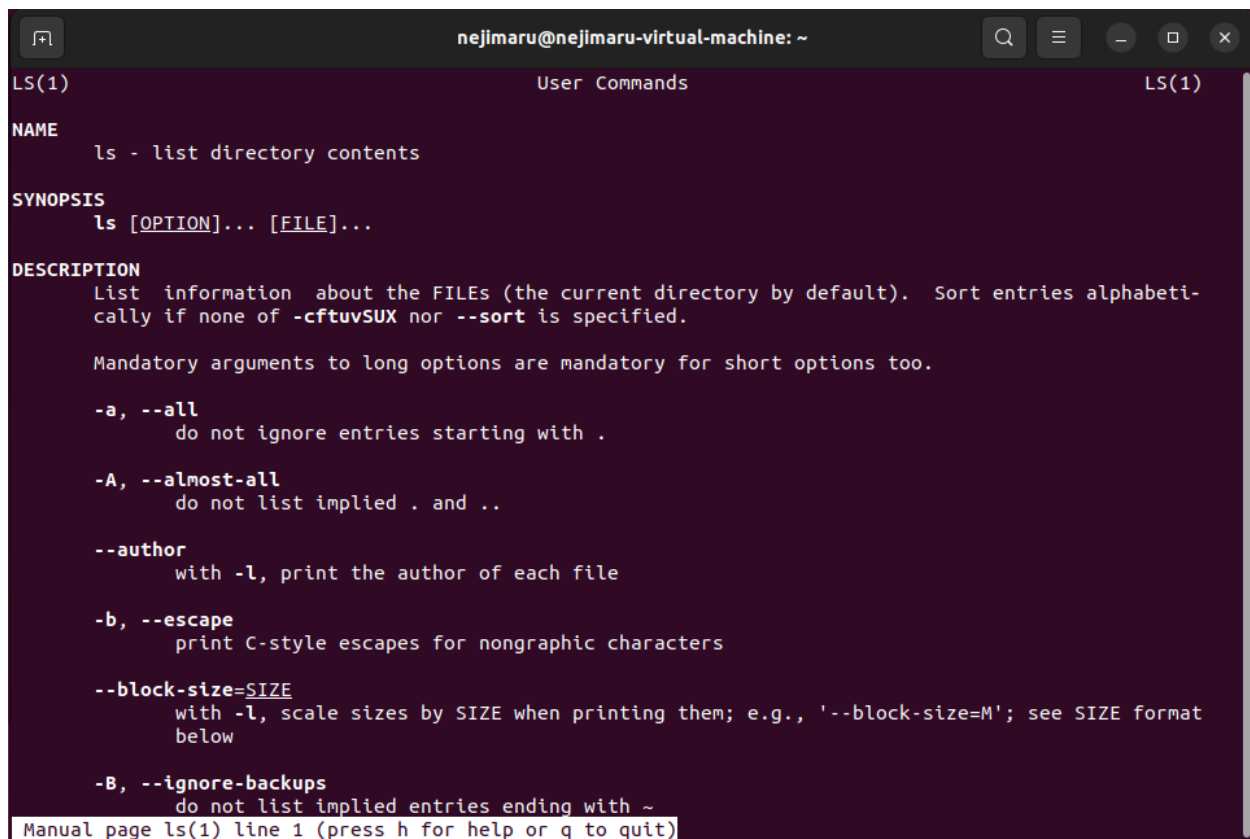


Groupe Shadows, réalisé Par Ibni-Yamine
19 septembre 2022

Job 1

- afficher le manuel de commande linux

Pour afficher le manuel des commandes linux, dans le terminal il faut entrer la commande `man` suivi de la commande que l'on veut afficher dans le manuel. il ici on affichera le manuel pour la commande `ls` :



```
nejimaru@nejimaru-virtual-machine: ~
LS(1) User Commands LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .
  -A, --almost-all
      do not list implied . and ..
  --author
      with -l, print the author of each file
  -b, --escape
      print C-style escapes for nongraphic characters
  --block-size=SIZE
      with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
  -B, --ignore-backups
      do not list implied entries ending with ~
Manual page ls(1) line 1 (press h for help or q to quit)
```

d'après le manuel, la commande `ls -a` nous permet d'afficher les fichiers et dossiers cachés dans un répertoire ciblé.

```
nejimaru@nejimaru-virtual-machine: ~  
nejimaru@nejimaru-virtual-machine:~$ pwd  
/home/nejimaru  
nejimaru@nejimaru-virtual-machine:~$ ls -a  
.          .bash_logout  .cache      .gnupg      .local      .profile    .ssh          Vidéos  
..         .bashrc       .config     Images      Modèles     Public      .sudo_as_admin_successful  
.bash_history Bureau        Documents   .lessht     Musique     snap        Téléchargements  
nejimaru@nejimaru-virtual-machine:~$
```

la commande `ls -a` nous affiche les dossier et fichier cachées dans notre répertoire `/home/nejimaru` .

Pour afficher les dossier et fichier présent dans le home sous forme de liste on utilise la commande `ls -l` .

```
nejimaru@nejimaru-virtual-machine:~$ ls -l  
total 36  
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Bureau  
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 19 10:54 Documents  
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Images  
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Modèles  
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Musique  
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Public  
drwx----- 3 nejimaru nejimaru 4096 sept. 18 19:14 snap  
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Téléchargements  
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Vidéos  
nejimaru@nejimaru-virtual-machine:~$
```

`ls -l` nous permettent d'afficher le type du fichier, les permissions d'accès, le nombre de liens physiques, le nom du propriétaire et du groupe, la taille en octets, et l'horodatage .

- Les options sont répertorié dans le manuel `man`, selon les commande comme pour la commande `ls` on y a ajouté l'option `-l`. Il y a plusieurs sortes d'options qui ont des actions différentes.
- les deux syntaxes principales d'écriture pour une option sont : d'entrer un tiret " - " suivi d'une lettre minuscule ou majuscule selon les options répertorié dans le manuel `man`, ou d'entrer "--" suivi d'un mot selon le manuel de la commande `man`
ex : `ls --all` fait la même chose que `ls -a` .

Job 2

il existe plusieurs commandes qui nous permettent d'afficher des fichiers texte dans le terminal, mais ces commandes font parfois des choses différentes.

la commande qui nous permet seulement de lire **cat** suivi du nom de notre fichier texte.
voici un exemple:

```
nejimaru@nejimaru-virtual-machine:~$ cat test
personne ne m'a interdit
que créer un fichier
texte juste pour
illustrer mon
exemple.
si vous lisez ceci, cela veut dire
que vous avez tout lu.
```

j'ai affiché le contenu de mon fichier **test** avec la commande **cat**, le fichier test a été lu dans son intégralité par le terminal.

ici nous allons afficher le contenu du fichier texte ".bashrc" dans le terminal,

```
nejimaru@nejimaru-virtual-machine:~$ 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

nejimaru@nejimaru-virtual-machine:~$
```

La commande **head** nous affiche par défaut les 10 premières lignes du fichier texte.

```
nejimaru@nejimaru-virtual-machine:~$ 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
nejimaru@nejimaru-virtual-machine:~$
```

La commande **tail** nous affiche par défaut les 10 dernières lignes de notre fichier texte .bashrc .

pour afficher les 20 premières lignes du fichier .bashrc on utilisera encore la commande head mais avec des options en plus :

```
-n, --lines=[-]NUM
    print the first NUM lines instead of the first 10; with the leading '-', print all
    but the last NUM lines of each file
```

d'après le manuel de la commande man pour la commande head, l'ajout de l'option -n accompagnés du nombre des premières lignes que l'on veut afficher.

```
nejimaru@nejimaru-virtual-machine:~$ 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
    *) ;;
    *) 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
nejimaru@nejimaru-virtual-machine:~$
```

Nous avons bien les 20 premières lignes du fichier texte “.bashrc” car, nous avons choisi le nombre 20 pour afficher les premières lignes. on peut aussi entrer **head -20 .bashrc** le résultat est le même.

nous pouvons faire la même chose pour les 20 dernières lignes , le manuel man de la commande tail nous fait utiliser la même option.

```
-n, --lines=[+]NUM
    output the last NUM lines, instead of the last 10; or use -n +NUM to output starting
    with line NUM
```

avec la commande et l'option **tail -n 20 .bashrc**, nous afficherons les 20 dernières ligne du fichier texte .bashrc

```
nejimaru@nejimaru-virtual-machine:~$ 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
nejimaru@nejimaru-virtual-machine:~$
```

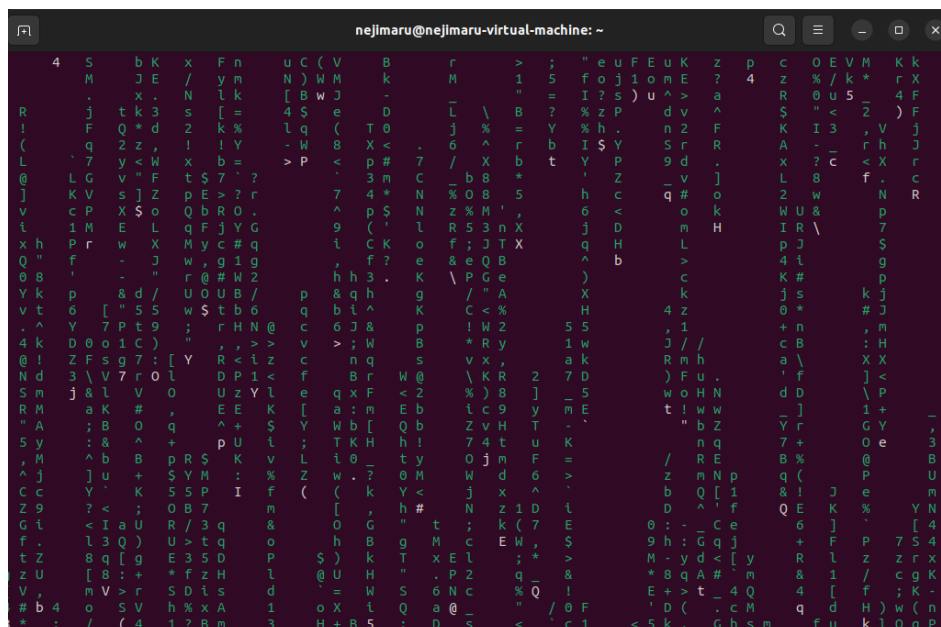
Nous avons bien les 20 dernières lignes du fichier .bashrc . nous pouvons même écrire “tail -20 .bashrc” .

Job 3

Nous allons installer les paquets de “cmatrix” avec la commande **sudo apt install cmatrix** .

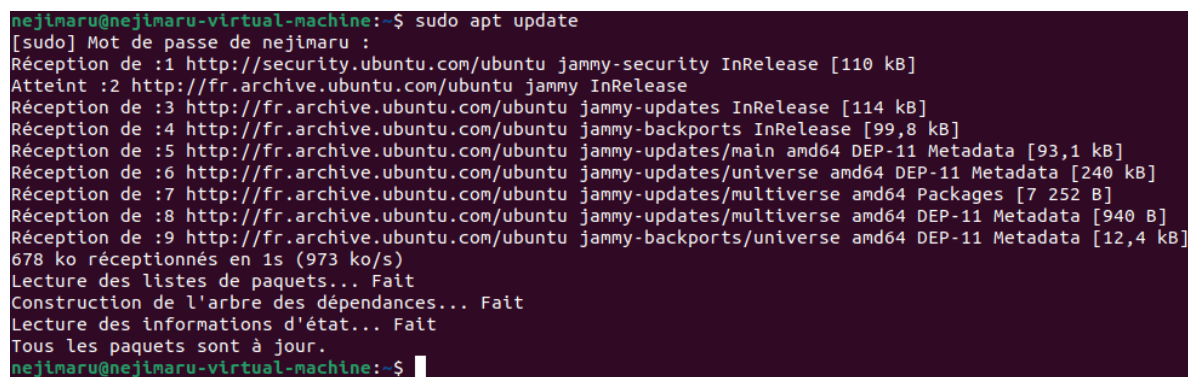
```
nejimaru@nejimaru-virtual-machine:~$ sudo apt install cmatrix
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
cmatrix est déjà la version la plus récente (2.0-3).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
nejimaru@nejimaru-virtual-machine:~$
```

Page 10 of 10

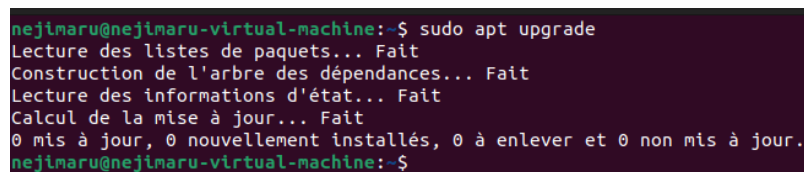


cmatrix est bien lancée, Bienvenue dans la matrice .

on met à jour le gestionnaire de paquets apt avec la commande **sudo apt update**



et pour mettre à jour tous les logiciels installés à leur dernière version, il faut utiliser la commande **sudo apt upgrade**



ici les logiciels sont déjà à jour .

nous allons installer google chrome depuis notre terminal avec **wget**.

Wget est un programme en ligne de commande non interactif de téléchargement de fichiers depuis le Web. Il supporte les protocoles HTTP, HTTPS et FTP ainsi que le téléchargement au travers des proxies HTTP.

Wget est installé par défaut dans les dernières versions d'Ubuntu et c'est le cas pour notre version.

il faudra entrer dans le terminal la commande :

wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb

```
nejimaru@nejimaru-virtual-machine:~/Documents$ wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
--2022-09-19 17:21:59-- https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
Résolution de dl.google.com (dl.google.com)... 142.250.201.14, 2a00:1450:4006:80e::200e
Connexion à dl.google.com (dl.google.com)|142.250.201.14|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 91239488 (87M) [application/x-debian-package]
Enregistre : 'google-chrome-stable_current_amd64.deb'

google-chrome-stable_current 100%[=====] 87,01M 26,1MB/s ds 3,4s
2022-09-19 17:22:02 (25,8 MB/s) - 'google-chrome-stable_current_amd64.deb' enregistré [91239488/91239488]
nejimaru@nejimaru-virtual-machine:~/Documents$
```

wget nous a télécharger les paquets de google chrome au format **.deb** dans le répertoire home.

et pour l'installer on entre la commande depuis home:

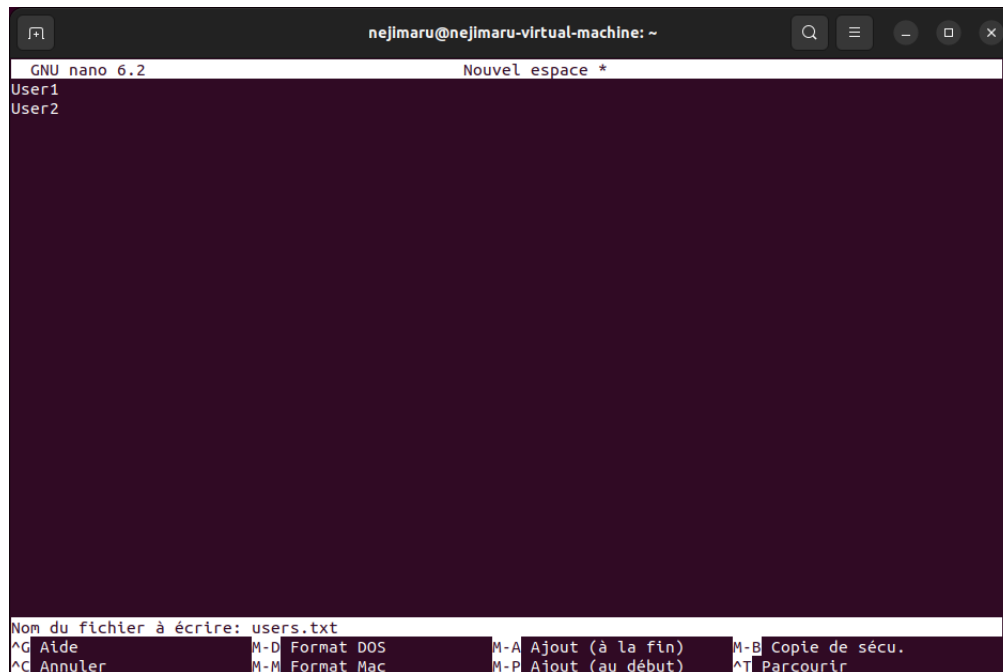
sudo dpkg -i google-chrome-stable_current_amd64.deb

```
nejimaru@nejimaru-virtual-machine:~$ sudo dpkg -i google-chrome-stable_current_amd64.deb
Sélection du paquet google-chrome-stable précédemment désélectionné.
(Lecture de la base de données... 175863 fichiers et répertoires déjà installés.
)
Préparation du dépaquetage de google-chrome-stable_current_amd64.deb ...
Dépaquetage de google-chrome-stable (105.0.5195.125-1) ...
Paramétrage de google-chrome-stable (105.0.5195.125-1) ...
update-alternatives: utilisation de « /usr/bin/google-chrome-stable » pour fournir « /usr/bin/x-www-browser » (x-www-browser) en mode automatique
update-alternatives: utilisation de « /usr/bin/google-chrome-stable » pour fournir « /usr/bin/gnome-www-browser » (gnome-www-browser) en mode automatique
update-alternatives: utilisation de « /usr/bin/google-chrome-stable » pour fournir « /usr/bin/google-chrome » (google-chrome) en mode automatique
Traitement des actions différées (« triggers ») pour mailcap (3.70+nmu1ubuntu1)
...
Traitement des actions différées (« triggers ») pour gnome-menus (3.36.0-1ubuntu3) ...
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.26-1ubuntu3) ...
Traitement des actions différées (« triggers ») pour man-db (2.10.2-1) ...
nejimaru@nejimaru-virtual-machine:~$
```

Google est bien installé, et on peut le lancer avec la commande **google-chrome-stable** .

Job 4

Il faut nous créer un fichier **users.txt** qui contiendra User1 et User2 séparé par un retour à la ligne. Je pourrais utiliser la commande **touch** users.txt pour créer mon fichier mais mais je vais faire autrement, j'utiliserai l'éditeur de texte **nano** avec la commande nano.

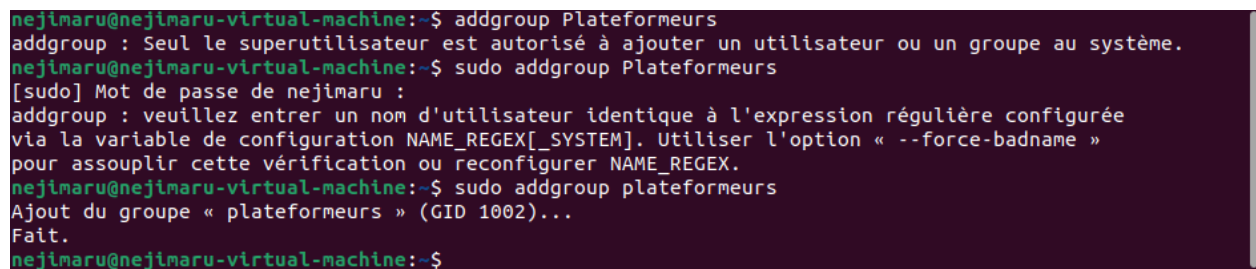


```
nejimaru@nejimaru-virtual-machine: ~
GNU nano 6.2                               Nouvel espace *
User1
User2

Nom du fichier à écrire: users.txt
^G Aide      ^M-D Format DOS  ^M-A Ajout (à la fin)  ^M-B Copie de sécu.
^C Annuler   ^M-M Format Mac  ^M-P Ajout (au début) ^M-T Parcourir
```

Lors de l'enregistrement du fichier, je lui ajoute l'extension **.txt** .

pour créer un groupe on utilise la commande **addgroup** suivi du nom de notre choix à mettre en paramètre notre groupe se nommera **plateformeurs** :



```
nejimaru@nejimaru-virtual-machine:~$ addgroup Plateformeurs
addgroup : Seul le superutilisateur est autorisé à ajouter un utilisateur ou un groupe au système.
nejimaru@nejimaru-virtual-machine:~$ sudo addgroup Plateformeurs
[sudo] Mot de passe de nejimaru :
addgroup : veuillez entrer un nom d'utilisateur identique à l'expression régulière configurée
via la variable de configuration NAME_REGEX[_SYSTEM]. Utiliser l'option « --force-badname »
pour assouplir cette vérification ou reconfigurer NAME_REGEX.
nejimaru@nejimaru-virtual-machine:~$ sudo addgroup plateformeurs
Ajout du groupe « plateformeurs » (GID 1002)...
Fait.
nejimaru@nejimaru-virtual-machine:~$
```

il faut créer un groupe en tant que **sudo** pour avoir le droit de créer un groupe avec les droits de superutilisateur sans en être un. ici le P majuscule me pose problème, j'ai donc nommé le groupe **plateformeurs**.

On va ajouter les utilisateurs **User1** et **User2** .

Pour ajouter des utilisateurs on utilise la commande **adduser** en **sudo** suivi du nom de notre utilisateur **user1** , lors de la création du compte on nous demande de créer un nouveau mot de passe et de le confirmer, on pourra mettre la première lettre de user1 en majuscule pour le choix du nom complet, après avoir validé les infos à entrer, on valide que nos infos entrées sont bien exactes avec **o** pour **oui** et voilà ! notre compte User1 est créé.

```
nejimaru@nejimaru-virtual-machine:~$ sudo adduser user1
Ajout de l'utilisateur « user1 » ...
Ajout du nouveau groupe « user1 » (1003) ...
Ajout du nouvel utilisateur « user1 » (1002) avec le groupe « user1 » ...
Création du répertoire personnel « /home/user1 »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
MOT DE PASSE INCORRECT : Le mot de passe comporte moins de 8 caractères
Retapez le nouveau mot de passe :
passwd : le mot de passe a été mis à jour avec succès
Modification des informations relatives à l'utilisateur user1
Entrez la nouvelle valeur ou « Entrée » pour conserver la valeur proposée
  Nom complet []: User1
  N° de bureau []:
  Téléphone professionnel []:
  Téléphone personnel []:
  Autre []:
Ces informations sont-elles correctes ? [0/n] o
nejimaru@nejimaru-virtual-machine:~$
```

on a fait pareil pour le user2.

Nous voulons ajouter **User2** au groupe **plateformeurs** .

nous allons utiliser la commande **usermod** pour modifier un compte d'utilisateur.

d'après le manuel **man usermod** , il faudrait utiliser l'option **-g**

```
-g, --gid GROUP
  Nom du groupe ou identifiant numérique du groupe de connexion initial de l'utilisateur. Le groupe doit exister.

  Tout fichier du répertoire personnel de l'utilisateur appartenant au groupe primaire précédent de l'utilisateur appartiendra à ce
  nouveau groupe.

  Le groupe propriétaire des fichiers en dehors du répertoire personnel de l'utilisateur doit être modifié manuellement.
```

Donc pour ajouter User2 dans le groupe plateformeurs, il faudra entrer dans le terminal:

usermod -g plateformeurs user2

on veut notre fichier "users.txt" dans un fichier "droits.txt" on utilisera la commande **cp**

```
nejimaru@nejimaru-virtual-machine:~$ cp users.txt droits.txt
nejimaru@nejimaru-virtual-machine:~$ ls
Bureau          google-chrome-stable_current_amd64.deb  Modèles  snap          Vidéos
Documents       groupes.txt                             Musique  Téléchargements
droits.txt      Images                                  Public   users.txt
nejimaru@nejimaru-virtual-machine:~$
```

la copie a bien fonctionné, j'ai fait la même chose pour le fichier **groupes.txt** .

Pour changer le propriétaire d'un fichier il faudra utiliser la commande **chown** .

On veut que User1 devienne le propriétaire du fichier droits.txt . voici la commande à entrer en **sudo**:

```
nejimaru@nejimaru-virtual-machine:~$ sudo chown user1 droits.txt
[sudo] Mot de passe de nejimaru :
nejimaru@nejimaru-virtual-machine:~$ ls -l
total 89152
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Bureau
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 19 17:37 Documents
-rw-rw-r-- 1 user1 nejimaru 12 sept. 20 14:57 droits.txt
-rw-rw-r-- 1 nejimaru nejimaru 91239488 sept. 10 01:11 google-chrome-stable_current_a
md64.deb
-rw-rw-r-- 1 nejimaru nejimaru 12 sept. 20 14:55 groupes.txt
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Images
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Modèles
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Musique
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Public
drwx----- 4 nejimaru nejimaru 4096 sept. 19 17:05 snap
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Téléchargements
-rw-rw-r-- 1 nejimaru nejimaru 12 sept. 20 10:27 users.txt
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Vidéos
nejimaru@nejimaru-virtual-machine:~$
```

sudo chown user1 droits.txt , user1 est devenu propriétaire du fichier droits.txt et on a pu le vérifier avec la commande **ls -l** mais user1 fait partie du groupe nejimaru .

on veut que user2 ai seulement le droit de lecture du fichier **droits.txt** .

La modification d'accès se fait avec la commande **chmod**.

Ici je vais attribuer les droits en chmod en absolu, c'est-à-dire avec les chiffres. pour cela il faut voir comment cela fonctionne.

Droits	Chiffre	Calcul
---	0	0 + 0 + 0
r--	4	4 + 0 + 0
-w-	2	0 + 2 + 0
--x	1	0 + 0 + 1
rw-	6	4 + 2 + 0
-wx	3	0 + 2 + 1
r-x	5	4 + 0 + 1
rwX	7	4 + 2 + 1

d (Directory) : indique si l'élément est un dossier ;

l (Link) : indique si l'élément est un lien (raccourci) ;

r (Read) : indique si on peut lire l'élément ;

w(Write) : indique si on peut modifier l'élément ;

x (eXecute) : si c'est un fichier, « x » indique qu'on peut l'exécuter. Ce n'est utile que pour les fichiers exécutables (programmes et scripts).

Si c'est un dossier, « x » indique qu'on peut le « traverser », c'est-à-dire qu'on peut voir les sous-dossiers qu'il contient si on a le droit de lecture dessus.



Ce schéma nous apprend comment déchiffrer les droits et à qui ils sont attribués.

dans notre cas, ce sera **chmod 764 droits.txt**, le propriétaire du fichier droits.txt est user1 donc on lui attribue les droit de lecture, écriture et exécution, ça correspond au chiffre **7** le deuxième chiffre correspond membres du groupe du propriétaire user1, ici **user1** appartient au groupe **nejimaru** on lui attribue l'écriture et la lecture, ça correspond au chiffre **6**, et pour finir les autres qui ne font pas partie du groupe du propriétaire, ils doivent avoir la possibilité de lecture, ça correspond au chiffre **4**

```
nejimaru@nejimaru-virtual-machine:~$ sudo chmod 764 droits.txt
nejimaru@nejimaru-virtual-machine:~$ ls -l
total 89152
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Bureau
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 19 17:37 Documents
-rwxrw-r-- 1 user1 nejimaru 12 sept. 20 14:57 droits.txt
-rw-rw-r-- 1 nejimaru nejimaru 91239488 sept. 10 01:11 google-chrome-stable_current_amd64.deb
-rw-rw-r-- 1 nejimaru nejimaru 12 sept. 20 20:55 groupes.txt
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Images
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Modèles
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Musique
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Public
drwx----- 4 nejimaru nejimaru 4096 sept. 19 17:05 snap
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Téléchargements
-rw-rw-r-- 1 nejimaru nejimaru 12 sept. 20 10:27 users.txt
drwxr-xr-x 2 nejimaru nejimaru 4096 sept. 18 19:14 Vidéos
nejimaru@nejimaru-virtual-machine:~$
```

“user2” fait partie du groupe “plateformeurs”, il ne fait donc pas partie du groupe “nejimaru” dont le propriétaire du fichier droits.txt “user1” est membre. user2 est un donc un utilisateur extérieur du groupe **nejimaru**, il fait donc partie des **autres**, user2 ne que **lire** le fichier droit.txt .

Maintenant nous allons changer les droits du fichier “groupes.txt” pour que les utilisateurs puissent accéder au fichier en lecture uniquement.

```
nejimaru@nejimaru-virtual-machine:~$ ls -l groupes.txt
-rw-rw-r-- 1 nejimaru nejimaru 12 sept. 20 20:55 groupes.txt
nejimaru@nejimaru-virtual-machine:~$
```

ici on peut voir que le propriétaire peut seulement lire et écrire **rw-**, que les groupe aussi **rw-** et les ne peuvent que lire **r--**, on a pas besoins de changer les droits de lecture pour les autre utilisateur car ils font pas partie du groupe **nejimaru**

et pour finir on va changer les droits du fichier pour que le groupe “Plateformeurs” puissent y accéder en lecture/écriture.

pour faire ça on va utiliser la commande **chgrp**.

Cette commande permet de changer le groupe d’un fichier. **user2** fait partie du groupe **plateformeurs**,

chgrp s'utilise exactement de la même manière que **chown** à la différence près qu'il affecte cette fois le groupe propriétaire d'un fichier.

la commande à entrer : **chgrp plateformeurs groupes.txt**

```
nejimaru@nejimaru-virtual-machine:~$ sudo chgrp plateformeurs groupes.txt
nejimaru@nejimaru-virtual-machine:~$ ls -l
total 89152
drwxr-xr-x 2 nejimaru nejimaru      4096 sept. 18 19:14 Bureau
drwxr-xr-x 2 nejimaru nejimaru      4096 sept. 19 17:37 Documents
-rwxrw-r-- 1 user1    nejimaru       12 sept. 20 14:57 droits.txt
-rw-rw-r-- 1 nejimaru nejimaru    91239488 sept. 10 01:11 google-chrome-stable_current_amd64.deb
-rw-rw-r-- 1 nejimaru plateformeurs  12 sept. 20 20:55 groupes.txt
drwxr-xr-x 2 nejimaru nejimaru      4096 sept. 18 19:14 Images
drwxr-xr-x 2 nejimaru nejimaru      4096 sept. 18 19:14 Modèles
drwxr-xr-x 2 nejimaru nejimaru      4096 sept. 18 19:14 Musique
drwxr-xr-x 2 nejimaru nejimaru      4096 sept. 18 19:14 Public
drwx----- 4 nejimaru nejimaru      4096 sept. 19 17:05 snap
drwxr-xr-x 2 nejimaru nejimaru      4096 sept. 18 19:14 Téléchargements
-rw-rw-r-- 1 nejimaru nejimaru       12 sept. 20 10:27 users.txt
drwxr-xr-x 2 nejimaru nejimaru      4096 sept. 18 19:14 Videos
```

Le fichier groupe.txt appartient à désormais au groupe **plateformeurs** et il appartient toujours à l'utilisateur **nejimaru**. **User2** fait partie du groupe **plateformeurs**, il bénéficie donc des droits sur le fichier lié à son groupe. **”rw-“** il peut lire et écrire sur le fichier groupe.txt.

Job 5

Dans mon terminal, la commande “**la**” est déjà un alias pour la commande “**ls -A**” qui a pour effet d’afficher les dossiers et fichiers de la ou se trouve notre répertoire y compris les fichiers cachés.

cet alias est écrit par défaut dans le fichier **.bashrc** , nous pouvons créer nos propre alias dans un fichier différent.

Pour éviter de modifier trop souvent le fichier sensible qu'est **.bashrc**, il est conseillé d'utiliser le fichier **.bash_aliases** Pour que celui-ci soit pris en compte, je n’ai pas de fichier **.bash_aliases** par défaut, alors je l’ai créé. il faut maintenant vérifier qu’il y a une liaison les fichier **.bashrc** et **.bash_aliases**, pour vérifier ça il faut le vérifier dans le fichier **.bashrc** (nano **.bashrc**)

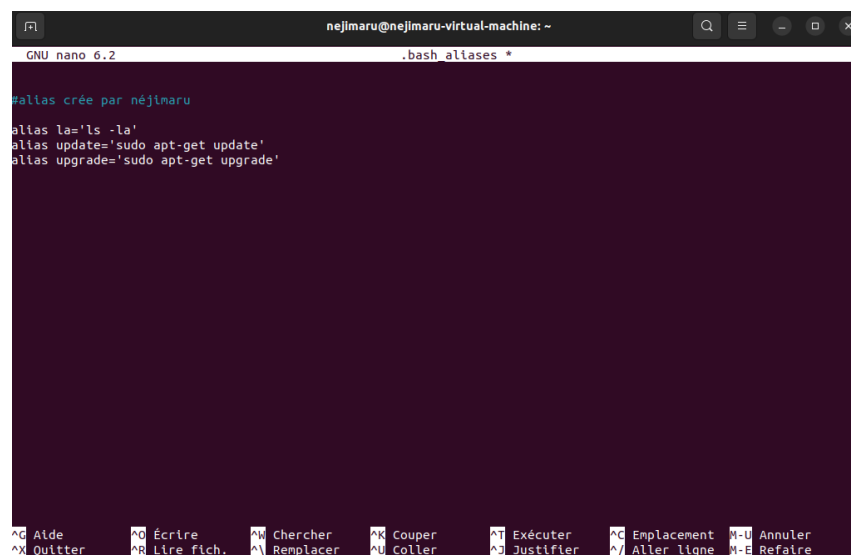
Depuis de **.bashrc**

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

on peut voir la condition du **.bashrc** qui nous laisse lancer nos alias depuis le **.bash_aliases**

allons créer nos alias :

Pour que nos alias soient pris en compte après ajout dans les fichiers **.bashrc** ou **.bash_aliases**, il vous faudra relancer votre terminal ou saisir la commande



```
nejimaru@nejimaru-virtual-machine: ~
GNU nano 6.2 .bash_aliases *

#alias crée par nejjimaru
alias la='ls -la'
alias update='sudo apt-get update'
alias upgrade='sudo apt-get upgrade'
```

Pour que nos alias soient pris en compte après ajout dans les fichiers **.bashrc** ou **.bash_aliases**, il vous faudra relancer notre terminal ou saisir la commande : **source ~/.bashrc**

```
nejimaru@nejimaru-virtual-machine:~$ la
total 89208
drwxr-x--- 17 nejimaru nejimaru      4096 sept. 21 14:17 .
drwxr-xr-x  6 root      root          4096 sept. 20 11:25 ..
-rw-rw-r--  1 nejimaru nejimaru      120 sept. 21 14:15 .bash_aliases
-rw-----  1 nejimaru nejimaru     3076 sept. 21 13:41 .bash_history
-rw-r--r--  1 nejimaru nejimaru      220 sept. 18 19:09 .bash_logout
-rw-r--r--  1 nejimaru nejimaru     3853 sept. 21 11:29 .bashrc
drwxr-xr-x  2 nejimaru nejimaru      4096 sept. 18 19:14 Bureau
drwx----- 11 nejimaru nejimaru      4096 sept. 19 17:49 .cache
drwx----- 14 nejimaru nejimaru      4096 sept. 20 20:48 .config
drwxr-xr-x  2 nejimaru nejimaru      4096 sept. 19 17:37 Documents
-rwxrw-r--  1 user1     nejimaru       12 sept. 20 14:57 droits.txt
drwx----- 2 nejimaru nejimaru      4096 sept. 19 17:27 .gnupg
-rw-rw-r--  1 nejimaru nejimaru    91239488 sept. 10 01:11 google-chrome-stable_current_amd64.deb
-rw-rw-r--  1 nejimaru plateformeurs  12 sept. 20 20:55 groupes.txt
drwxr-xr-x  2 nejimaru nejimaru      4096 sept. 18 19:14 Images
-rw-----  1 nejimaru nejimaru       20 sept. 21 10:41 .lessht
drwx-----  3 nejimaru nejimaru      4096 sept. 18 19:14 .local
drwxr-xr-x  2 nejimaru nejimaru      4096 sept. 18 19:14 Modèles
drwxr-xr-x  2 nejimaru nejimaru      4096 sept. 18 19:14 Musique
drwx-----  3 nejimaru nejimaru      4096 sept. 19 17:49 .pkl
-rw-r--r--  1 nejimaru nejimaru       808 sept. 21 11:28 .profile
drwxr-xr-x  2 nejimaru nejimaru      4096 sept. 18 19:14 Public
drwx-----  4 nejimaru nejimaru      4096 sept. 19 17:05 snap
drwx----- 2 nejimaru nejimaru      4096 sept. 18 19:23 .ssh
-rw-r--r--  1 nejimaru nejimaru         0 sept. 19 09:38 .sudo_as_admin_successful
drwxr-xr-x  2 nejimaru nejimaru      4096 sept. 18 19:14 Téléchargements
-rw-rw-r--  1 nejimaru nejimaru       12 sept. 20 10:27 users.txt
drwxr-xr-x  2 nejimaru nejimaru      4096 sept. 18 19:14 Vidéos
nejimaru@nejimaru-virtual-machine:~$
```

```
nejimaru@nejimaru-virtual-machine:~$ update
[sudo] Mot de passe de nejimaru :
Atteint :1 http://fr.archive.ubuntu.com/ubuntu jammy InRelease
Atteint :2 https://dl.google.com/linux/chrome/deb stable InRelease
Réception de :3 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Réception de :4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Réception de :5 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease [99,8 kB]
Réception de :6 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [584 kB]
Réception de :7 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [315 kB]
Réception de :8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [13,1 kB]
Réception de :9 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [93,0 kB]
Réception de :10 http://fr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [417 kB]
Réception de :11 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [12,2 kB]
Réception de :12 http://fr.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [274 kB]
Réception de :13 http://fr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [240 kB]
Réception de :14 http://fr.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [7 256 B]
Réception de :15 http://fr.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Réception de :16 http://fr.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12,3 kB]
2 293 ko réceptionnés en 1s (1 737 ko/s)
Lecture des listes de paquets... Fait
nejimaru@nejimaru-virtual-machine:~$ upgrade
```

```
nejimaru@nejimaru-virtual-machine:~$ upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
 evolution-data-server evolution-data-server-common fonts-opensymbol libcamel-1.2-63 libebook-1.2-10
 libebook-1.2-20 libebook-contacts-1.2-3 libecal-2.0-1 libedata-book-1.2-26 libedata-cal-2.0-1
 libedataserver-1.2-26 libedataserverui-1.2-3 linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04
 linux-image-generic-hwe-22.04
0 mis à jour, 0 nouvellement installés, 0 à enlever et 15 non mis à jour.
```

les commandes alias **la**, **update**, **upgrade** fonctionnent,

mais il est préférable de ne pas inclure ajouter **sudo** dans la déclaration de l'alias si on est débutant ou pour nous empêcher de déclarer des privilèges élevés.

Je vais déclarer une variable d'environnement qui se nommera "**USER**" et qui sera égale à mon nom d'utilisateur **nejimaru**

je rentre ma variable depuis mon fichier **.bash_aliases** (on peut déclarer nos variables directement dans le terminal mais j'ai décidé de le faire dans le fichier **.bash_aliases**)
on déclare une variable avec **export**

```
nejimaru@nejimaru-virtual-machine: ~  
GNU nano 6.2 .bash_aliases  
  
#alias crée par néjimaru  
alias la='ls -la'  
alias update='sudo apt-get update'  
alias upgrade='sudo apt-get upgrade'  
  
export USER=nejimaru
```

je met a jours les modifications de mon **.bash_aliases** qui est liée à mon **.bashrc** avec la commande : **source ~/.bashrc**

on peut vérifier si notre variable d'environnement est bien créé avec la commande , **echo** permet d'afficher le contenu d'un variable et d'afficher aussi d'afficher du texte écrit à la suite de la commande.

echo + \$nom_de_notre_variable

```
nejimaru@nejimaru-virtual-machine:~$ echo $USER  
nejimaru
```

ici ma variable est bien égale à **nejimaru** qui est mon nom d'utilisateur.

Pour voir toutes les variables d'environnement on utilise la commande **env** .

on va maintenant ajouter au Path de notre variable **USER** le chemin **"/home/nejimaru/Bureau"**
on peut voir notre variable **PATH** avec la commande **env**

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin  
GDMSESSION=ubuntu  
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus  
_=/usr/bin/env
```

on va garder seulement le chemin :

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
```

notre ajout se dera a la fin

mais on ajoutera ":" après bin puis notre chemin **/home/nejimaru/Bureau** .

```
nejimaru@nejimaru-virtual-machine:~$ export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin:/home/nejimaru/Bureau
```


ATTENTION de bien respecter la syntaxe au risque d'écraser les chemins et de rendre les commandes **ls** inutilisables.

on fait un `echo` pour vérifier si notre chemin a bien été ajouté .

```
nejimaru@nejimaru-virtual-machine:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/nejimaru/Bureau
nejimaru@nejimaru-virtual-machine:~$
```

il est aussi visible avec la commande **env**

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/nejimaru/Bureau
```

notre chemin a bien été ajouté !

Job 7

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 :

avant tout il nous faut connaître les fonctions de chaque caractère:

le caractère ">" fait la redirection, si le fichier existe il est effacé et remplacé, et le fichier n'existe pas il est créé, mais il sera vide.

le caractère ">>" nous permet de créer un fichier et d'entrer du contenu avec la commande **echo**

```
nejimaru@nejimaru-virtual-machine:~$ echo "Je suis votre fichier texte" >> une_commande.txt
nejimaru@nejimaru-virtual-machine:~$ ls
Bureau      google-chrome-stable_current_and64.deb  Modèles  snap      users.txt
Documents   groupes.txt                             Musique  Téléchargements  Vidéos
droits.txt  Images                                  Public   une_commande.txt
nejimaru@nejimaru-virtual-machine:~$ cat une_commande.txt
Je suis votre fichier texte
```

nous devons maintenant afficher le nombre de lignes contenu dans le fichier **sources.list** qui se trouve dans **/etc/apt/sources.list** , et stocker le résultat dans le fichier **nb_lignes.txt**

je vais utiliser la commande **wc** (word count) qui permet d'afficher le nombre de ligne, de mot et de caractère.

"<" permet de récupérer le résultat d'une commande "**wc**" liée à une cible "notre fichier source.list"

“>” copie et redirige le résultat des commandes précédente et les enregistre.

```
nejimaru@nejimaru-virtual-machine:~$ wc < /etc/apt/sources.list > nb_lignes.txt
```

```
nejimaru@nejimaru-virtual-machine:~$ cat nb_lignes.txt
50 334 2824
```

il y a 50 lignes dans mon fichier sources.list qui se trouve dans **/etc/apt/sources.list**

On veut maintenant afficher le contenu du fichier source.list et l’enregistrer dans un autre fichier appelé “save_sources”

on affiche la cible qui est **sources.list** avec la commande **cat**, on redirige le résultat de l’affichage **cat** avec “>” vers le fichier **save_sources** “notre fichier save_sources n’existe vail être créé”

```
nejimaru@nejimaru-virtual-machine:~$ cat /etc/apt/sources.list > save_sources | ls
Bureau      google-chrome-stable_current_amd64.deb  Modèles     Public      Téléchargements  Vidéos
Documents   groupes.txt                             Musique     save_sources  une_commande.txt
droits.txt  Images                                  nb_lignes.txt snap         users.txt
nejimaru@nejimaru-virtual-machine:~$
```

notre fichier save_sources est bien créé

```
nejimaru@nejimaru-virtual-machine:~$ cat save_sources
#deb cdrom:[Ubuntu 22.04.1 LTS _Jammy Jellyfish_ - Release amd64 (20220809.1)]/ jammy main restricted

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://fr.archive.ubuntu.com/ubuntu/ jammy main restricted
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://fr.archive.ubuntu.com/ubuntu/ jammy-updates main restricted
# deb-src http://fr.archive.ubuntu.com/ubuntu/ jammy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
```

je n’affiche pas tout le contenu mais le fichier **save_sources** à bien le même contenu que le fichier **sources.list** .

Pour finir on veut rechercher des fichiers commençant par “.” tout en cherchant le mot **alias** qui sera utilisé depuis **un** fichier

on va utiliser la commande **grep** qui nous permet de rechercher des mot dans un fichier. Nous allons rechercher le mot **alias** et à la suite le nom du fichier dans lequel on veut rechercher notre mot. Nous voulons chercher le mot alias dans tous les fichiers qui contiennent alias et notre fichier commence par “.” on va ajouter une astérisque ***** qui signifiera tous les fichiers existant commençant par “.” la commande complète : **grep alias .*** mais j’ai rajouté l’option **-n** pour voir à quel ligne du fichier se trouve le mot alias.

```
nejimaru@nejimaru-virtual-machine:~$ grep -n alias .*
grep: .: est un dossier
grep: ./: est un dossier
.bash_aliases:3:#alias crée par nejimaru
.bash_aliases:5:alias la='ls -la'
.bash_aliases:6:alias update='sudo apt-get update'
.bash_aliases:7:alias upgrade='sudo apt-get upgrade'
.bash_history:238:nano bash_aliases
.bash_history:254:nano .bash_aliases
.bash_history:257:nano .bash_aliases
.bash_history:259:nano .bash_aliases
.bash_history:264:nano .bash_aliases
.bash_history:267:nano .bash_aliases
.bash_history:275:nano .bash_aliases
.bash_history:281:nano .bash_aliases
.bash_history:284:nano .bash_aliases
.bash_history:310:nano .bash_aliases
.bash_history:312:nano .bash_aliases
.bash_history:326:nano .bash_aliases
.bash_history:337:nano .bash_aliases
.bash_history:465:grep -rn "alias"
.bash_history:467:find | grep -rn "alias"
.bash_history:469:grep -rn "alias" | d.*
.bash_history:482:find -name ".*" | grep -rn 'alias'
.bash_history:483:find -name ".*" || grep -rn 'alias'
.bash_history:484:find -name ".*" | grep -rn 'alias'
.bash_history:485:find -name ".*" || grep -rn 'alias'
.bash_history:486:find -name ".*" -f || grep -rn 'alias'
.bash_history:487:find -name ".*" || grep -rn 'alias'
.bash_history:495:find /home/nejimaru/ -name ".*" -type f | grep -rn 'alias'
.bash_history:497:find /home/nejimaru/ -name ".*" -type f | grep -rn 'alias'
.bash_history:499:grep alias .*
.bash_history:500:grep -rn 'alias'
.bash_history:501:find /home/nejimaru/ -name ".*" -type f | grep alias .*
.bash_history:502:find /home/nejimaru/ -name ".*" -type f && grep alias .*
.bash_history:503:find /home/nejimaru/ -name ".*" -type f & grep alias .*
.bash_history:504:find /home/nejimaru/ -name ".*" -type f | grep -rn 'alias'
.bash_history:505:find /home/nejimaru/ -name ".*" -type f & grep alias .*
.bash_history:506:grep alias .bashrc
.bash_history:507:find /home/nejimaru/ -name ".*" -type f | grep alias .bashrc
.bash_history:508:find /home/nejimaru/ -name ".*" -type f | grep alias .*
.bash_history:510:find /home/nejimaru/ -name ".*" -type f | grep alias .bashrc
.bash_history:513:find /home/nejimaru/ -name ".*" -type f | grep alias .bashrc
.bashrc:75:# enable color support of ls and also add handy aliases
.bashrc:78:alias ls='ls --color=auto'
```

on trouve bien tous les **alias** dans tous les fichiers commençant par “.”

Pour aller plus loin . . .

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

on doit Installer la commande tree avec la commande **sudo apt-get install tree**

```
nejimaru@nejimaru-virtual-machine:~$ tree
.
├── Bureau
├── Documents
│   └── test.txt
├── droits.txt
├── google-chrome-stable_current_amd64.deb
├── groupes.txt
├── Images
│   ├── 1040737.png
│   └── LePathDoNotDel.txt
├── Modèles
├── Musique
├── nb_lignes.txt
├── Public
├── save_sources
├── snap
│   ├── firefox
│   │   ├── 1810
│   │   ├── 1860
│   │   ├── common
│   │   └── current -> 1860
│   └── snapd-desktop-integration
│       ├── 14
│       │   ├── Bureau
│       │   ├── Documents
│       │   ├── Images
│       │   ├── Modèles
│       │   ├── Musique
│       │   ├── Public
│       │   ├── Téléchargements
│       │   └── Vidéos
│       ├── common
│       └── current -> 14
├── Téléchargements
├── une_commande.txt
├── users.txt
└── Vidéos

26 directories, 10 files
nejimaru@nejimaru-virtual-machine:~$
```

tree est bien installé et fonctionne correctement.

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**"

on lance tree avec la commande **tree**, on copie et on redirige sont son contenu avec le chevron ">" vers le fichier **tree.save**

```
nejimaru@nejimaru-virtual-machine:~$ tree > tree.save
nejimaru@nejimaru-virtual-machine:~$ cat tree.save
.
├── Bureau
├── Documents
│   └── test.txt
├── droits.txt
├── google-chrome-stable_current_amd64.deb
├── groupes.txt
├── Images
│   ├── 1040737.png
│   └── LePathDoNotDel.txt
├── Modèles
├── Musique
├── nb_lignes.txt
├── Public
├── save_sources
├── snap
│   ├── firefox
│   │   ├── 1810
│   │   ├── 1860
│   │   ├── common
│   │   └── current -> 1860
│   └── snapd-desktop-integration
│       ├── 14
│       │   ├── Bureau
│       │   ├── Documents
│       │   ├── Images
│       │   ├── Modèles
│       │   ├── Musique
│       │   ├── Public
│       │   ├── Téléchargements
│       │   └── Vidéos
│       ├── common
│       └── current -> 14
├── Téléchargements
├── tree.save
├── une_commande.txt
├── users.txt
└── Vidéos

26 directories, 11 files
nejimaru@nejimaru-virtual-machine:~$
```

On veut maintenant lister les éléments présents dans le dossier courant est utilisé directement le résultat de notre première commande pour compter le nombre d'éléments trouvés pour lister les éléments présents dans le dossier courant, c'est la commande **ls**, le commande 'l' qui veut dire 'et' qui va nous permet d'exécuter la première commande et d'entrée une autre commande, tout ça en une seul ligne de commande, et pour compter le nombre de fichier présent c'est la commande **wc** qui permet d'afficher le nombre de ligne, de mot et de caractère.

dans notre cas avec il sera accompagné de la commande **ls**, donc il comptera le nombre de fichier présent en rapport avec **ls**. on veut qu'il compte seulement le nombre de fichiers et pas le reste. on utilisera **wc** avec l'option **-l** pour compter seulement les fichiers (sources : **man** de **wc**).

```
nejimaru@nejimaru-virtual-machine:~$ ls
Bureau      google-chrome-stable_current_amd64.deb  Modèles      Public      Téléchargements  users.txt
Documents   groupes.txt                             Musique       save_sources tree.save         Videos
droits.txt  Images                                  nb_lignes.txt snap          une_commande.txt

nejimaru@nejimaru-virtual-machine:~$ ls | wc -l
17
nejimaru@nejimaru-virtual-machine:~$
```

on a bien 17 fichiers dans notre répertoire courant.

Lancer une commande pour update et upgrade.

```
nejimaru@nejimaru-virtual-machine:~$ update
[sudo] Mot de passe de nejimaru :
Atteint :1 http://fr.archive.ubuntu.com/ubuntu jammy InRelease
Réception de :2 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Atteint :3 https://dl.google.com/linux/chrome/deb stable InRelease
Réception de :4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Réception de :5 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease [99,8 kB]
Réception de :6 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [322 kB]
Réception de :7 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [597 kB]
Réception de :8 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [140 kB]
Réception de :9 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [92,8 kB]
Réception de :10 http://fr.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [278 kB]
Réception de :11 http://fr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [422 kB]
Réception de :12 http://fr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [247 kB]
Réception de :13 http://fr.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [944 B]
Réception de :14 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [13,1 kB]
Réception de :15 http://fr.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12,3 kB]
Réception de :16 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [12,1 kB]
2 461 ko réceptionnés en 2s (1 345 ko/s)
Lecture des listes de paquets... Fait
nejimaru@nejimaru-virtual-machine:~$ upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
  gzip libnss-systemd libpam-systemd libpython3-stdlib libpython3.10 libpython3.10-minimal
  libpython3.10-stdlib libspeechd2 libsystemd0 libudev1 python3 python3-gdbm python3-minimal
  python3-speechd python3.10 python3.10-minimal speech-dispatcher speech-dispatcher-audio-plugins
  speech-dispatcher-espeak-ng systemd systemd-oemd systemd-sysv systemd-timesyncd udev xwayland
0 mis à jour, 0 nouvellement installés, 0 à enlever et 25 non mis à jour.
nejimaru@nejimaru-virtual-machine:~$
```

tout fonctionne !!! pas d'échec lors de mon update, j'ai pu faire un upgrade avec mes alias **update** et **upgrade** .