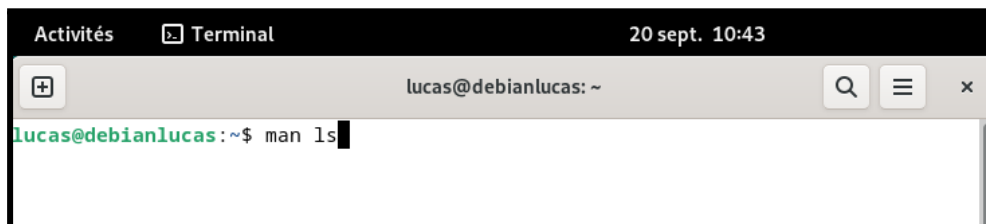


PROJET : SHELL

<u>Job 1</u>	page 1
<u>Job 2</u>	page 3
<u>Job 3</u>	page 4
<u>Job 4</u>	page 5
<u>Job 5</u>	page 7
<u>Job 6</u>	page 10
<u>Job 7</u>	page 11
<u>Pour aller plus loin</u>	page 14
<u>Glossaire</u>	page 16

Job 1 :


→ Afin d'afficher le manuel de la commande ls il suffit d'ouvrir le terminal (shell) et d'entrer la commande **man ls**



```
lucas@debianlucas:~$ man ls
```

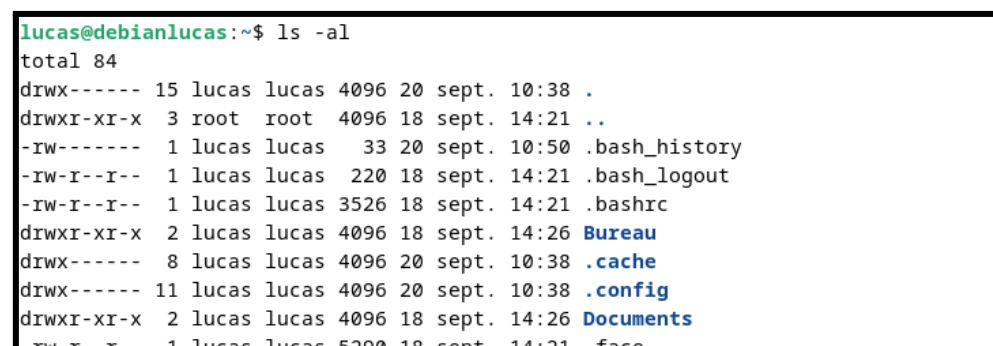
→ Pour afficher les fichiers cachés du home, il faut entrer : **ls -a**

Le tilde (~) avant le \$ signifie qu'on se trouve dans le répertoire /home de l'utilisateur en cours.



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

→ Pour afficher les fichiers cachés plus leurs droits il faut entrer : **ls -la**



```
lucas@debianlucas:~$ ls -la
total 84
drwx----- 15 lucas lucas 4096 20 sept. 10:38 .
drwxr-xr-x  3 root  root  4096 18 sept. 14:21 ..
-rw-----  1 lucas lucas   33 20 sept. 10:50 .bash_history
-rw-r--r--  1 lucas lucas  220 18 sept. 14:21 .bash_logout
-rw-r--r--  1 lucas lucas 3526 18 sept. 14:21 .bashrc
drwxr-xr-x  2 lucas lucas 4096 18 sept. 14:26 Bureau
drwx-----  8 lucas lucas 4096 20 sept. 10:38 .cache
drwx----- 11 lucas lucas 4096 20 sept. 10:38 .config
drwxr-xr-x  2 lucas lucas 4096 18 sept. 14:26 Documents
-rw-r--r--  1 lucas lucas 5290 18 sept. 14:21 .face
```

La commande **man** permet d'ouvrir le manuel.

La commande **ls** permet d'afficher le contenu des répertoires.

la commande **l** permet à la commande **ls** d'ouvrir une liste détaillée dont les informations sur leurs droits.

La commande **a** permet d'afficher tous les fichiers.

Les commandes **-a** et **-l** peuvent être “fusionnées” permettant d’entrer la commande **-la** ou **-al** pour obtenir le même résultat. Associé à la commande **ls** nous affichons tous les fichiers cachés ainsi que leurs droits.

→ Pour ajouter une option il suffit d’écrire la commande voulu comme **-l** pour lister ou **-d** pour afficher les répertoires précédé d’un espace.

→ Les deux syntaxes principales pour les options sont - (tiret du six) suivie du raccourci de la commande (a, d, l, ...) ou alors **double tiret** suivie de la commande entière (all, directory, ...)

Job 2 :

→ Pour lire un document (“.bashrc” par exemple) nous devons entrer la commande **cat /home/lucas/.bashrc**

→ Pour afficher les 10 premières lignes de du fichier “.bashrc” nous devons entrer la commande **head /home/lucas/.bashrc**

→ Pour afficher les 10 dernières lignes de du fichier “.bashrc” nous devons entrer la commande **tail /home/lucas/.bashrc**

→ Pour afficher les 20 premières lignes de du fichier “.bashrc” nous devons entrer la commande **head -n 20 /home/lucas/.bashrc**

→ Pour afficher les 20 dernières lignes de du fichier “.bashrc” nous devons entrer la commande **tail -n 20 /home/lucas/.bashrc**

```
lucas@debianlucas:~$ head -n 20 /home/lucas/.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
```

/home/lucas/.bashrc est le chemin du fichier que nous voulons lire.

La commande **cat** permet de lire un document.

La commande **head** permet de lire les premières lignes d'un document, sans options, seules les 10 premières lignes seront lues.

La commande **tail** permet de lire les dernières lignes d'un document, sans options, seules les 10 dernières lignes seront lues.

L'option **-n 20** précédé de head ou tail permet de définir le nombre de ligne à lire (20 lignes ici par exemple)

Job 3 :

→ Pour installer le paquet “cmatrix” il faut entrer la commande **sudo apt-get install cmatrix**, un mot de passe est demandé.

```
lucas@debianlucas:~$ sudo apt-get 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.
```

La commande **sudo** permet d'entrer une commande en “super-utilisateur”.

→ **cmatrix** est un utilitaire permettant d'ouvrir la matrix sur shell.



```
lucas@debianlucas: ~
2 Y V F      6 p S [ v f ' k F _ \ R K ! O O S _ 2 q $ $
p N K r      8 J p V ' Y m < ; l w g \ X ? < m " r
d G ' d ( > s 3 k o 9 R 8 W 1 Q ! e ^ u X \
I 0 $ t 0 6 Z g M P N ] w b G I 8 m ` z * 8 M
c , f v v J u l u I ' " G ` c k a 0 * m * L A
D A M @ F S > % 3 r d z ` H # ] W g X Q w Z ?
& h u n q = n E C / f B : D g 1 v j m * h >
q * 4 H s d W / d 0 i \ c m V g $ H - 6 b
q 9 a Y $ l f W + & n x A l 0 @ h - p f Y m
```

→ Afin de mettre à jour le gestionnaire de paquets (**apt-get upgrade**) il suffit d'écrire la commande **sudo apt-get update**.

→ Une fois à jour il faut utiliser la commande **sudo apt-get upgrade** pour mettre à jour ses logiciels.

```
lucas@debianlucas:~$ sudo apt-get update
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://deb.debian.org/debian bookworm-updates InRelease
Réception de :3 http://security.debian.org/debian-security bookworm-security InR
elease [48,0 kB]
48,0 ko réceptionnés en 1s (73,5 ko/s)
Lecture des listes de paquets... Fait
lucas@debianlucas:~$ sudo apt-get 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
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```

→ Afin de redémarrer, il faut entrer la commande **sudo reboot**, pour l'arrêter il faut utiliser la commande **sudo poweroff**.

Job 4 :

→ Pour créer le fichier users.txt il faut entrer la commande **nano users.txt**

nano étant la commande pour créer un fichier texte éditable.

La commande touch permet de créer un fichier txt, la commande nano permet de le créer et de le modifier.

```
GNU nano 7.2          users.txt
User1
User2
```

→ Pour créer le groupe, on utilise la commande

sudo groupadd Plateformeurs

→ Pour créer les utilisateurs 1 et 2, il faut entrer la commande

sudo adduser User1 et **sudo adduser User2** puis de leurs configurer un mot de passe.

→ Pour ajouter User2 au groupe Plateformeurs il faut entrer la commande

sudo adduser User2 Plateformeurs.

(User2 étant l'utilisateur que l'on veut déplacer Plateformeurs étant le groupe ciblé)

→ Afin de copier le fichier, on utilise la commande **cp users.txt droits.txt**

(Users.txt qui est la source, Droits.txt qui est la cible), de même pour le fichier groupes.txt, on entre **cp users.txt groupes.txt**

On peut vérifier le bon fonctionnement de la manipulation en entrant la commande **cat droits.txt**

```
lucas@debianlucas:~$ cat droits.txt
User1
User2
```

→ Pour changer le propriétaire d'un fichier on entre la commande

sudo chown User1 droits.txt (chown pour *Change owner*)

En vérifiant les droits déjà présent sur le fichier droits.txt avec la commande **ls -l droits.txt** le User2 possède uniquement la lecture.

```
root@debianlucas:/home/lucas# ls -l droits.txt
-rw-r--r-- 1 User1 root 12 20 sept. 16:16 droits.txt
```

(signifié par le 3ème “r” (Lecture) et non suivie d’un “w” (Écriture))

→ Dans le cas ou il aurait les droits d’écriture il faudrait entrer la commande **sudo chmod o-w droits.txt** (ou **chmod** est la commande pour modifier les droits, **o** pour tous les autres, **-** pour retirer, **w** pour l’écriture)

→ De même pour groupes.txt, les utilisateurs sont déjà en lecture uniquement mais on pourrait entrer la commande **sudo chmod o-w groupes.txt** si ce n’était pas le cas.

→ Pour modifier les droits de Plateformeur sur le fichier groupes.txt, il faut commencer par changer le groupe du fichier avec la commande **sudo chgrp Plateformeurs groupes.txt** (**chgrp** pour changement de groupe que j’ai attribués à Plateformeurs pour le fichier)

Ensuite il faut entrer la commande **sudo chmod g+w groupes.txt**

Si elle n’avait ni les droits de lecture ni les droits d’écriture il faudrait entrer

sudo chmod g+wr groupes.txt

(**chmod** pour modifier, **g** pour groupe, **+** pour ajouter et **w/r** pour Écrire/Lire)

```
lucas@debianlucas:~$ ls -l groupes.txt
-rw-rw-r-- 1 root Plateformeurs 12 20 sept. 16:20 groupes.txt
```

Job 5 :

Notre premier objectif est de créer un alias sur Linux. Un alias est un raccourci d'une commande qui serait trop longue à écrire ou qu'il faudrait utiliser de trop nombreuses fois.

Afin de créer un alias, il faut utiliser la commande **alias** suivie du nom qu'on veut lui donner (**la** dans notre cas) suivie d'un **=** et enfin de la commande qu'on veut lui attribuer (**ls -la** dans notre cas)

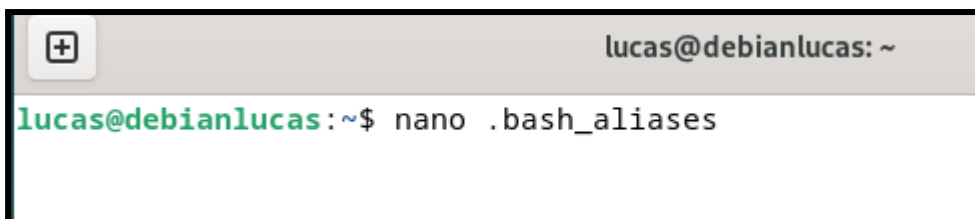
→ La commande est donc **alias la="ls -la"**

Pour rendre cette commande permanente, deux possibilités s'offrent à nous :
Editer le fichier `.bashrc` et inscrire notre commande en dessous de "`#some more aliases`"

```
# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'
```

(les commandes présentent ci dessus et précédé d'un # sont des commandes d'exemples qui ne sont pas prise en compte (du fait du #))

Pour éviter de modifier le fichier `.bashrc` (qui est le script de shell c'est à dire qu'il automatisera les commandes inscrite dans celui-ci à chaque lancement) il existe une seconde possibilité qui consiste à créer un fichier **`.bash_aliases`** à l'aide de la commande **nano .bash_aliases**
Une fois dans l'éditeur ouvert il suffit d'entrer notre alias et de sauvegarder.



```
lucas@debianlucas: ~$ nano .bash_aliases
```


→ On procède de la même manière pour apt-get update et apt-get upgrade avec les commandes : **alias update="sudo apt-get update"** et **alias upgrade="sudo apt-get upgrade"**

```
lucas@debianlucas: ~  
GNU nano 7.2 .bash_aliases *  
alias la="ls -la"  
alias update="sudo apt-get update"  
alias upgrade="sudo apt-get upgrade"
```

→ Une variable d'environnement permet de modifier le comportement des programmes (Comme la langue par exemple)
Pour ajouter la variable d'environnement demandé, il faut entrer la commande **export USER=lucas**
La commande export permet que la commande soit valable pour tous les autres environnements.

→ Afin de mettre à jour le shell, on va le rafraîchir à l'aide de la commande source. On utilise donc la commande **source .bashrc**

```
lucas@debianlucas: ~  
lucas@debianlucas:~$ export USER=lucas  
lucas@debianlucas:~$ echo $USER  
lucas  
lucas@debianlucas:~$ source .bashrc  
lucas@debianlucas:~$
```

→ Pour afficher les variables environnement, il nous faut utiliser la commande **printenv**

```
lucas@debianlucas: ~  
lucas@debianlucas:~$ printenv  
SHELL=/bin/bash  
SESSION_MANAGER=local/debianlucas:~/tmp/.ICE-unix/1457,u  
E-unix/1457  
QT_ACCESSIBILITY=1  
COLORTERM=truecolor
```

→ Pour modifier notre PATH et y ajouter un chemin, on utilise la commande ***PATH=\$PATH:/home/lucas/Bureau*** . Afin de le rendre permanent, on écrit cette commande dans le fichier .bashrc

```
lucas@debianlucas:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
lucas@debianlucas:~$ source .bashrc
lucas@debianlucas:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/lucas/Bureau
```

Le \$ de garder la valeur de l'élément PATH, les deux points superposés permettent de séparer l'ancienne valeur que nous gardons et notre nouveau chemin.

Le PATH cherche systématiquement les fichiers à utiliser lorsque l'on entre le nom d'une commande.

Job 6 :

→ Afin de décompresser le fichier Ghost in the Shell, on utilise la commande

tar -xvf Documents/Ghost\ in\ the\ Shell.tar

(tar -xvf est la commande de désarchivage et les slashes inversés servent à conserver les espaces). Le symbole ~ présent avant le \$ indique que nous nous trouvons déjà dans le dossier lucas, il n'est donc pas nécessaire de préciser les parties antérieures du chemin (soit /home/lucas)

```
lucas@debianlucas:~$ tar -xvf Documents/Ghost\ in\ the\ Shell.tar
._Ghost in the Shell.pdf
```

Pour vérifier le bon fonctionnement de notre commande, on peut utiliser la commande ***ls***, si le fichier "Ghost in the Shell.pdf" est présent, le désarchivage à fonctionner.

```
lucas@debianlucas:~$ ls
Bureau      'Ghost in the Shell.pdf'  Modèles  Téléchargements
Documents   groupes.txt               Musique   users.txt
droits.txt  Images                    Public    Vidéos
```

Job 7 :

→ Afin de créer notre fichier texte avec avec un message à l'intérieur il faut entrer la commande

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

echo permet d'afficher un message ou d'afficher le résultat d'une autre commande.

">" permet de rediriger vers une sortie (dans notre cas, le fichier "une_commande.txt").

Si le fichier n'existe pas, celui-ci est automatiquement créé.

```
lucas@debianlucas:~$ echo "Je suis votre fichier texte" > une_commande.txt
lucas@debianlucas:~$ cat une_commande.txt
"Je suis votre fichier texte"
```

(On vérifie avec la commande **cat une_commande.txt**)

→ Pour compter le nombre de lignes présentes le fichier il faut entrer la commande

cat /etc/apt/sources.list | wc -l > nb_lignes.txt

cat sert principalement à lire du texte, **wc** nous permet de compter et **-l** précise que ce sont les lignes qu'il faut compter (On aurait pu compter les mots en utilisant **wc -w** par exemple) Encore une fois, nous redirigeons notre résultat avec **>** vers notre fichier texte. Enfin, **|** (la barre verticale) permet de lier des commandes entre elles

```
lucas@debianlucas:~$ cat /etc/apt/sources.list | wc -l > nb_lignes.txt
lucas@debianlucas:~$ cat nb_lignes.txt
18
```

(On vérifie avec la commande **cat nb_lignes.txt**)

→ Afin d'afficher et de copier sources.list vers save_source, on utilise :

cat /etc/apt/sources.list | tee save_sources

On utilise **cat** pour afficher notre sources.list.

Notre | (barre verticale) nous permet de lier les commandes.

tee permet de lire et d'écrire (vers save_sources dans notre cas)

```
lucas@debianlucas:~$ cat /etc/apt/sources.list | tee save_sources
#deb cdrom:[Debian GNU/Linux 12.1.0 _Bookworm_ - Official amd64 NETIN
mware 20230722-10:48]/ bookworm main non-free-firmware
deb http://deb.debian.org/debian/ bookworm main non-free-firmware
```

→ Afin de lister tous les mots "alias" dans nos fichiers commençant par "." il faut entrer la commande

grep -r --include=".*" "alias"

Notre **-r** indique à grep de rechercher récursivement dans les sous-répertoires
--include=".*" indique que seuls les fichiers dont le nom commence par "." sont cherché.

L'astérisque "*" signifie que cela sera valable pour tous les fichiers du répertoire

"alias" est le mot quel'on recherche à l'intérieur de ces fichiers.

```
lucas@debianlucas:~$ grep -r --include=".*" "alias"
.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'
```

→ Enfin, pour lier les commandes entre elle, on utilise “;” qui permet de mettre les commandes à la suite. On peut également utiliser “&&” qui permet aussi de les mettre à la suite mais qui les lancent uniquement dans le cas ou la précédente à fonctionner.

echo “Je suis votre fichier texte” > une_commande.txt ; cat /etc/apt/sources.list | wc -l > nb_lignes.txt ; cat /etc/apt/sources.list | tee save_sources ; grep -r --include=".*" "alias"

```
lucas@debianlucas:~$ echo "Je suis votre fichier texte" > une_commande.txt ; cat
/etc/apt/sources.list | wc -l > nb_lignes.txt ; cat /etc/apt/sources.list | tee
save_sources ; grep -r --include=".*" "alias"
#deb cdrom:[Debian GNU/Linux 12.1.0 _Bookworm_ - Official amd64 NETINST with fir
mware 20230722-10:48]/ bookworm main non-free-firmware

deb http://deb.debian.org/debian/ bookworm main non-free-firmware
deb-src http://deb.debian.org/debian/ bookworm main non-free-firmware
```

```
.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'
```

```
lucas@debianlucas:~$ ls
Bureau          groupes.txt     nb_lignes.txt  tree.save
Documents       Images         Public         une_commande.txt
droits.txt      Modèles        save_sources   users.txt
'Ghost in the Shell.pdf' Musique         Téléchargements Vidéos
```

Pour aller plus loin

→ Pour installer tree on utilise la commande

sudo apt-get install tree

&& permet de mettre plusieurs commandes à la suite si les précédentes commandes ont fonctionné

→ Pour lancer la commande tree en arrière-plan et afficher l'arborescence de / puis l'enregistrer dans "tree.save" on utilise la commande :

tree / > tree.save &

"tree /" affiche l'arborescence de "/" (les fichiers racine) et redirige grâce à ">" dans le fichier "tree.save"

& permet d'effectuer la commande en arrière plan.

→ Pour lister les éléments présents dans le dossier courant et compter le nombre d'éléments trouvés on utilise la commande :

(ls -A | wc -l)

ls -A permet d'afficher les fichiers du dossier courant
wc -l compte le nombre de ligne

→ Enfin, pour installer update et upgrade, on utilise les commandes

sudo apt-get update && sudo apt-get upgrade

sudo apt-get update pour mettre à jour la liste des paquets et

sudo apt-get upgrade pour lancer la mise à jour des paquets.

Grâce à “&&”, la commande ne sera exécutée que si la commande précédente (**sudo apt-get update**) réussit.

→ La ligne de commande à écrire pour pouvoir réaliser toutes ces tâches est donc :

sudo apt-get install tree && (tree / > tree.save &) && (ls -A | wc -l) && sudo apt-get update && sudo apt-get upgrade

```
lucas@debianlucas:~$ sudo apt-get install tree && (tree / > tree.save &) && (ls -A | wc -l) && sudo apt-get update && sudo apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
tree est déjà la version la plus récente (2.1.0-1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
32
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
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
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```

```
lucas@debianlucas:~$ ls
```

Bureau

groupes.txt

nb_lignes.txt

tree.save

Glossaire

man	Afficher le manuel
ls	Afficher contenu d'un répertoire
-a	Afficher tous les fichiers (dont ceux cachés)
-l	Afficher le detail des fichiers dont les droits
cat	Lire un document
head	Affiche les 10 premières lignes d'un fichier
tail	Affiche les 10 dernières lignes d'un fichier
-n xx	Affiche les xx lignes d'un fichier
sudo	Entrer une commande en super user
apt-get install	Installer logiciel ou paquet
apt-get update	Mettre à jour le gestionnaire de paquets
apt-get upgrade	Mettre à jour les logicielles
reboot	Redémarrer la machine
poweroff	Eteindre la machine
nano	Créer un fichier texte
groupadd	Créer un groupe
adduser	Créer un utilisateur
cp	Copier un fichier
chown	Changer le propriétaire
chmod u/g/o/a r/w/x	Modifier les droits Utilisateur/Groupe/Autres/Tout le monde Lecture/Écriture/Exécution
chgrp	Changer le groupe
alias	Créer un alias (raccourcis de commande)
printenv	Afficher les variables environnements
PATH=	Modifie le PATH
tar	Permet de désarchiver

echo	Permet d'afficher