

Job 01:

Pour afficher le manuel de la commande ls, on utilise la commande “man”, pour “manual”:
man ls

On obtient alors une description de la commande ainsi que les options que l'on peut lui donner.

Pour afficher les fichiers cachés du /home, j'ouvre un terminal, et je remonte d'abord dans l'arborescence jusqu'à être dans le /home/, soit en utilisant cd .. (la commande cd signifiant “change directory” et les deux points référant le répertoire au-dessus du répertoire dans lequel on se trouve), soit en indiquant directement le répertoire dans lequel on veut se rendre, ici en utilisant cd /home/

Si on avait voulu se rendre à la racine de l'arborescence, on aurait fait cd /

Une fois que je suis dans mon répertoire /home, j'utilise la commande ls pour lister les éléments qui s'y trouvent.

Je peux alors utiliser plusieurs options, comme -a pour afficher les fichiers cachés, donc ici: ls -a

Ou même utiliser l'option -l pour afficher les éléments sous forme de liste avec les droits de chaque élément, donc ici: ls -l

Et enfin, je peux combiner les deux options: ls -a -l, mais je peux aussi abréger davantage en liant les deux options: ls -al (pratique lorsque l'on utilise plusieurs options à la commande ls, comme -l, -a, -h qui rend le poids de chaque élément plus lisible, -t qui organise les éléments par rapport à la dernière modification qu'on a faite dessus)

Donc: ajouter des options à une commande se fait avec un ou deux tirets (dépendant de l'option et de la façon dont on veut l'écrire)

Les deux syntaxes “principales” (bien que je pense qu'on peut en compter trois) sont: la syntaxe “courte”, en utilisant un tiret simple et une lettre unique, comme par exemple: ls -a

Ou la syntaxe “longue”, en utilisant un double tiret et un ou plusieurs mots complets: ls --all

Je pense qu'on peut ajouter à ces deux-là une autre syntaxe, en abrégant encore davantage, par exemple: ls -a -l -t -h (souvent utilisée) devient ls -alth, ls -a -l -t -h -R (option récursive, qui va donc exécuter un ls dans les sous-dossiers, puis les dossier encore en-dessous, etc) devient ls -althR, etc.

Job 02:

La commande cat permet d'afficher le contenu d'un fichier directement dans le Terminal: pour afficher le contenu du fichier “Hockey” dans lequel j'ai écrit “Nous avons gagné 2-0 contre Cergy hier soir”, qui se situe dans le répertoire /home/{utilisateur}/Bureau/, j'exécute: cat /home/{utilisateur}/Bureau/Hockey, qui me retourne: Nous avons gagné 2-0 contre Cergy hier soir

Je peux, sinon, utiliser cat en étant directement dans le bon répertoire avec cd /home/{utilisateur}/Bureau puis cat Hockey

La commande head permet d'afficher (par défaut) les 10 premières lignes d'un fichier. Je peux lui passer -n en option suivi d'un numéro pour afficher les {numéro} premières lignes du fichier.

Ainsi: head .bashrc (en se trouvant à la racine) affiche les 10 premières lignes de .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
```

Et: 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
```

La commande tail permet d'afficher, par défaut, les 10 dernières lignes d'un fichier:

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
```

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

```

Job 03:

Ne voulant pas installer cmatrix sur mon système Debian 12 personnel, je crée une VM et l'y installe.

```
sudo apt install cmatrix
```

Cette commande va chercher, avec le gestionnaire de paquets apt, dans des sources différentes spécifiées de base par l'OS et que l'on peut modifier, un paquet cmatrix à installer.

Pour mettre à jour son gestionnaire de paquets et ses logiciels, on utilise:

sudo (pour "superuser do", donc "exécuter en tant que superutilisateur, pour ça il nous faudra les droits de superutilisateur) apt update

```

merkava@mantak:~$ sudo apt update
Réception de :1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Atteint :4 http://download.virtualbox.org/virtualbox/debian bookworm InRelease
Réception de :6 http://repository.spotify.com stable InRelease [3 316 B]
Atteint :7 https://deb.opera.com/opera-stable stable InRelease
Err :6 http://repository.spotify.com stable InRelease
  Les signatures suivantes n'ont pas pu être vérifiées car la clé publique n'est pas disponible : NO
_PUBKEY 7A3A762FAFD4A51F
Atteint :5 https://deb.opera.com/opera stable InRelease

```

Qui va rechercher dans des sources prédéfinies si de nouvelles mises à jour de paquets sont disponibles pour nos paquets déjà installés (notez que j'ai installé spotify sur mon ordinateur et donc ajouté spotify à la liste des sources, mais qu'une erreur s'affiche: la source ne met pas de clé publique permettant l'authenticité de la potentielle mise à jour, donc apt update va l'ignorer).

Puis `sudo apt upgrade` pour installer les mises à jour que `sudo apt update` vient de repérer et mettre à jour les logiciels.

```
merkava@mantak:~$ sudo apt 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.
```

Pas de mise à jour faite ici, tout est déjà à jour, félicitations à moi-même.

La commande `wget` permet de télécharger un fichier sur le web, en utilisant les protocoles `http`, `https` ou encore `ftp`.

Sur une page comme `google.com`, sans fichier à transférer, `wget` téléchargera le code html de la page, et ici l'enregistrera en tant qu'`index.html` dans le dossier dans lequel on se trouve:

```
merkava@mantak:~$ wget google.com
--2023-09-20 16:38:43-- http://google.com/
Résolution de google.com (google.com)... 142.251.37.238, 2a00:1450:4006:813::200e
Connexion à google.com (google.com)|142.251.37.238|:80... connecté.
requête HTTP transmise, en attente de la réponse... 301 Moved Permanently
Emplacement : http://www.google.com/ [suivant]
--2023-09-20 16:38:43-- http://www.google.com/
Résolution de www.google.com (www.google.com)... 172.217.21.4, 2a00:1450:4006:800::2004
Connexion à www.google.com (www.google.com)|172.217.21.4|:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : non indiqué [text/html]
Sauvegarde en : « index.html »

index.html          [ <=> ] 22,22K --.-KB/s ds 0,009s

2023-09-20 16:38:43 (2,34 MB/s) - « index.html » sauvegardé [22756]
```

`index.html` se retrouve donc dans mon répertoire utilisateur `merkava`

```
merkava@mantak:~$ ls
Bureau      Musique      Téléchargements
Documents   Oracle_VM_VirtualBox_Extension_Pack-7.0.10.vbox-extpack  Vidéos
Images      Public       'VirtualBox VMs'
index.html  rtl88x2bu
Modèles     snap

merkava@mantak:~$ cat index.html
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="fr"><head><r
="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/go
gleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="ft1e
J5i0aQ">(function(){var _g={kEI:'8wMLZZOJDbythbIPpPmo8A8',kEXPI:'0,1365467,207,4804,2316
129120,1578491,16114,28684,22431,1361,12311,17588,4998,17075,38444,2872,2891,3926,7828,60
22,2614,13491,230,20583,1,1,2,32894,26723,27023,6651,7596,1,42154,2,39761,5679,1020,25049
6258,23418,1252,40029,17666,2006,8155,23350,873,19634,7,1922,9779,42459,20199,5796,4,1443
5,8377,8048,10912,2306,3097,3030,15816,1804,12748,8263,1826,4255,6005,2172,5249,6564,1635
,7875,3838,7914,12742,3,7,2,5400,5213623,108,2,195,5994770,96,2803118,3311,141,795,19736
452,254,17,14,6,4,6,5,6,4,22,12,1,10,22,20721486,2210067,578,4042528,16672,20282,4402,74
```

`sudo reboot` pour redémarrer l'ordinateur (ou `shutdown -r`, moins intuitif)

`sudo shutdown` pour l'éteindre

Job 04:

Pour créer un fichier texte, j'utilise nano (installé par défaut, le plus simple et rapide d'utilisation, bien que vim soit préférable pour la suite de l'apprentissage de Linux et de l'administration système):

`nano users.txt`

Puis j'écris User1, retour à la ligne, User2

```
GNU nano 7.2                                users *
User1
User2
```

J'enregistre, je quitte, et voilà

Les prochaines opérations se feront sur ma VM Debian 12 et non sur ma propre configuration:

Pour ajouter un groupe d'utilisateurs: `sudo groupadd {NOM DU GROUPE}`

Ici: `sudo groupadd Plateformeurs`

On vérifie que le groupe soit bien ajouté avec: `cat /etc/group | grep Plateformeurs`

On a alors plusieurs informations: `nom_groupe:mdp_groupe:ID_groupe:utilisateurs_groupe`

Donc, le nom du groupe soit Plateformeurs, le mdp du groupe (crypté si présent) donc ici x car il n'y a rien, l'ID du groupe donc ici 1001, et les utilisateurs dans le groupe donc ici rien
Soit: `Plateformeurs:x:1001:`

Pour créer un utilisateur, on utilise `sudo useradd -m` (option qui permet de lui ajouter un répertoire personnel dans le home, avec les permissions qui vont avec) `User1`

Pour lui assigner un mot de passe, on exécute la commande: `sudo passwd User1`

Puis on rentre le mot de passe souhaité, ici (à ne pas faire): `u1pass`

Même chose pour `User2`

Pour ajouter `User2` au groupe `Plateformeurs`: `sudo usermod -aG Plateformeurs User2`

La commande `usermod` permet de modifier le fichier des comptes utilisateurs

L'option `-a` couplée à l'option `G` permet de append (c'est-à-dire rajouter) un utilisateur à un groupe.

On vérifie: `cat /etc/group | grep Plateformeurs` :

```
Plateformeurs:x:1001:User2
```

Je copie ensuite `users.txt` dans un fichier `droits.txt` puis `groupes.txt` avec:

`cp users.txt droits.txt | cp users.txt groupes.txt`

On vérifie avec `ls`

```
droits.txt  groupes.txt  users.txt
```

Je fais un `ls -l` pour voir plusieurs choses: le propriétaire de chaque fichier, le groupe propriétaire de chaque fichier, ainsi que les droits du propriétaire, du groupe propriétaire, et du reste des utilisateurs sur le fichier:

```
-rw-r--r-- 1 merkava merkava 12 20 sept. 18:40 droits.txt
-rw-r--r-- 1 merkava merkava 12 20 sept. 18:40 groupes.txt
-rw-r--r-- 1 merkava merkava 12 20 sept. 16:56 users.txt
```

On voit ici que merkava est le propriétaire, que merkava est le groupe propriétaire (normal, par défaut l'utilisateur primaire de la machine est assigné à un groupe du même nom, et c'est l'utilisateur merkava qui a créé ces trois fichiers)

Pour changer le propriétaire, on utilise la commande `chown`: `chown User1 droits.txt`

Pour permettre à User2 (et tous les utilisateurs qui ne sont ni dans le groupe merkava, ni l'User1) de lire ce fichier uniquement, on effectue un `sudo chmod 774` (correspondant donc à un `rw-rwx-r`, c'est à dire à l'exécution des commandes `sudo chmod ug+rwx droits.txt` && `sudo chmod o+r`) `droits.txt`

Pour que les utilisateurs autres que merkava et ceux dans le groupe merkava aient accès en lecture uniquement au fichier `groupes.txt`, on utilise de nouveau la commande `chmod`:

`chmod 754` (ou `774`, si l'on veut donner les pleins droits également au groupe propriétaire):

`chmod 754 groupes.txt`

Enfin, pour permettre à Plateformeurs d'avoir accès au fichier `groupes.txt` en lecture/écriture (et aussi exécution, pourquoi pas), on exécute: `chgrp Plateformeurs groupes.txt`

Puis on exécute un `chmod 774` (donc `rw-rwx-r`): `chmod 774 groupes.txt`

Résultat final:

```
-rwxrwxr-- 1 User1  merkava      12 20 sept. 18:40 droits.txt
-rwxrwxr-- 1 merkava Plateformeurs 12 20 sept. 18:40 groupes.txt
-rw-r--r-- 1 merkava merkava      12 20 sept. 16:56 users.txt
```

Notons ici que je mets, par habitude, les droits d'exécution, mais je pourrais ne pas les mettre, les 7 dans les `chmod` deviendraient donc des 6.

Mais alors: pourquoi des chiffres (question rhétorique, je suppose que vous le savez mais je me permets de le préciser) ? C'est en fait la traduction binaire des droits que l'on veut mettre. Supposons que l'on ait un chiffre codé sur 3 bits, avec le premier bit (tout à droite) étant la permission `x`, le deuxième `w`, le troisième `r`: en mettant tous ces bits à 1, donc en donnant toutes les permissions, on a donc en binaire 1 1 1, ce qui donne, en décimal: $1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 7$

Si l'on veut donner uniquement la permission `r`, donc read, on passe donc le troisième bit à 1 et les deux autres à 0: en binaire 1 0 0, en décimal $1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 4$

Ainsi, le `chmod` au complet est codé sur 3 fois (pour les trois types de personnes: le propriétaire, le groupe propriétaire, et les autres) 3 bits, donc 9 bits.

Job 05:

Pour ajouter temporairement un alias, on peut exécuter la commande: `alias {alias désiré}="{commande visée}"`

Donc par exemple: `alias la="ls -la"`

Pour l'ajouter de manière permanente, cependant, il faut changer le fichier .bashrc à la racine de notre arborescence

Ce fichier est un script exécuté à la connexion d'un utilisateur, qui contient des informations sur la configuration du Terminal.

(On pourrait, en temps normal, ajouter les aliases dans un fichier différent de .bashrc, pour améliorer le confort de lecture, mais ici on va se contenter de ça)

Alors: nano .bashrc

Voici donc ce qu'on a (en plus du contenu de base) dans notre fichier .bashrc:

```
#aliases pour le job05
alias la='ls -la'
alias update='sudo apt update'
alias upgrade='sudo apt upgrade'
alias upgdrd='sudo apt update && sudo apt upgrade'
```

Pour la variable USER:

```
export USER='merkava'
```

On enregistre, on ferme nano.

Pour recharger le fichier, on utilise la commande: source .bashrc

On s'assure que le USER fonctionne bien: echo \$USER

On s'assure que les alias fonctionnent bien: la

```
total 17976
drwx----- 21 merkava merkava    4096 20 sept. 19:47 .
drwxr-xr-x  5 root    root        4096 20 sept. 18:04 ..
-rw-----  1 merkava merkava    5297 20 sept. 01:28 .bash_history
-rw-r--r--  1 merkava merkava     220 17 sept. 20:21 .bash_logout
-rw-r--r--  1 merkava merkava    3705 20 sept. 19:47 .bashrc
drwxr-xr-x  2 merkava merkava    4096 17 sept. 20:24 Bureau
drwx----- 22 merkava merkava    4096 20 sept. 15:10 .cache
drwx----- 19 merkava merkava    4096 20 sept. 16:07 .config
drwxr-xr-x  4 merkava merkava    4096 20 sept. 16:48 Documents
-rw-r--r--  1 merkava merkava    5290 17 sept. 20:21 .face
lrwxrwxrwx  1 merkava merkava      5 17 sept. 20:21 face.icon -> face
```

Pour afficher les variables d'environnement: commande env

Pour modifier le PATH, on rouvre .bashrc avec nano, puis on ajoute à la fin:

```
export PATH=$PATH:/home/merkava/Bureau
```

Les : servent ici à indiquer que l'on veut "append" quelque-chose, c'est-à-dire le rajouter à la fin

Un echo \$PATH nous donne alors:

```
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/snap/bin:/home/merkava/Bureau/
```



```

merkava@mantak:~$ env
SHELL=/bin/bash
SESSION_MANAGER=local/mantak:@/tmp/.ICE-unix/2025,unix/mantak:/tmp/.ICE-unix/2025
QT_ACCESSIBILITY=1
COLORTERM=truecolor
SSH_AGENT_LAUNCHER=openssh
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=gnome
GTK_MODULES=gail:atk-bridge
PWD=/home/merkava
LOGNAME=merkava
XDG_SESSION_DESKTOP=gnome
XDG_SESSION_TYPE=wayland
SYSTEMD_EXEC_PID=2079
XAUTHORITY=/run/user/1000/.mutter-Xwaylandauth.4Q3BB2
GDM_LANG=fr_FR.UTF-8
HOME=/home/merkava
USERNAME=merkava
IM_CONFIG_PHASE=1
LANG=fr_FR.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;0
1:mi=00:su=37;41:sg=30;43:ca=00:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01
;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31
:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01
;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:
.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz
=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;3
1:*.esd=01;31:*.avif=01;35:*.img=01;35:*.ineg=01;35:*.ming=01;35:*.mineg=01;35:*.gif=01;

```

Job 06:

Pour télécharger le fichier, je le fais en interface graphique, avec Opera (le télécharger par le Terminal requiert énormément de choses, étant donné que Google demande une authentification pour le document, il faudrait donc créer un cookie d'authentification avec wget, en plus de bypass la double authentification par un moyen que j'ignore, c'est bien au-dessus de moi pour le moment).

Je peux ensuite l'extraire avec tar -xzvf (ici x signifie extract, z signifie unzip sous entendu décompresser, v signifie verbose c'est-à-dire afficher au fur et à mesure le détail des éléments extraits, et f signifie que c'est bien un fichier que l'on veut extraire).

Donc:

```
tar -xzvf Ghost\ in\ the\ Shell\.
```

(les antislash sont ici pour signifier que l'espace est à prendre en compte pour le nom du fichier et non pour la commande en elle-même.

```

._Ghost in the Shell.pdf
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.quarantine » pour l'en-tête étendu a été ignoré
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.provenance » pour l'en-tête étendu a été ignoré
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.macl » pour l'en-tête étendu a été ignoré
Ghost in the Shell.pdf

```

On ouvre ensuite le pdf en graphique avec la commande evince Ghost\ in\ the\ Shell\.