



SHELL



Ubuntu

*École La Plateforme,
Établissement d'enseignement supérieur à Marseille,
Bachelor IT 1A, Programmation informatique.*



Sommaire

Sommaire	2
Job 01	3
Fichier de documentation	3
Questions	3
Job 02	4
Fichier de documentation	4
Job 03	5
Fichier de documentation	5
Job 04	6
Fichier de documentation	6
Job 05	8
Fichier de documentation	8
Job 06	9
Fichier de documentation	9
Job 07	10
Fichier de documentation	10
Pour aller plus loin...	12
Fichier de documentation	12
Bonus :)	14
Fichier de documentation	14
Questions	16

Job 01

Fichier de documentation

Afficher le manuel de la commande ls

La commande : `man ls`, donne le manuel de la commande `ls`.

Afficher les fichiers cachés du home de votre utilisateur

La commande : `ls`, permet de lister les éléments non cachés du dossier en cours.

La commande : `ls -a`, permet d'afficher tous les éléments ainsi que les éléments cachés.

La commande : `ls -d .*`, permet d'afficher que les fichiers et dossiers cachés.

La commande : `ls -d */`, permet d'afficher que les dossiers cachés.

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

La commande : `ls -al`, permet d'afficher les fichiers cachés ainsi que les informations sur les droits sous forme de liste.

Permissions pouvant être appliquées sur les fichiers/dossiers :

- `r` (pour `read`), droit de lire le fichier/dossier.
- `w` (pour `write`), droit de modifier le fichier/dossier.
- `x` (pour `execute`), droit d'exécuter le fichier.
- `-` (pour `rien`), aucun droit sur le fichier/dossier.

Questions

Comment ajouter des options à une commande ?

- Le premier mot tapé est une commande.
- Les lettres tapées après un tiret (tiret du six) et les mots tapés après deux tirets sont des options (le reste constitue les paramètres).

NB : il existe des commandes sans paramètres, d'autres sans options et même certaines qui prennent une commande en paramètre.

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

- **Le simple tiret** : elle consiste à précéder d'un tiret une lettre représentant une option. Il est également possible avec cette convention de regrouper plusieurs options dans un seul argument. Par exemple, la commande `ls -a -l` est identique à la commande `ls -al`.
- **Le double tiret** : elle permet de corriger le manque de lisibilité du simple tiret. Souvent ces deux conventions coexistent, par contre il n'est pas possible de combiner plusieurs options en une seule.

Job 02

Fichier de documentation

Lisez un fichier en utilisant une commande qui permet seulement de lire

La commande : `cat [fichier]`, permet d'afficher le contenu d'un fichier dans la sortie du terminal.

Par exemple : `cat .bashrc`, permet de lire le contenu du fichier `.bashrc`, dans la sortie du terminal.

Afficher les 10 premières lignes du fichier ".bashrc"

La commande : `head [fichier]`, permet d'afficher les dix premières lignes d'un fichier dans la sortie du terminal. Par exemple : `head .bashrc`, permet d'afficher les dix premières lignes du fichier `.bashrc`, dans la sortie du terminal.

Afficher les 10 dernières lignes du fichier ".bashrc"

La commande : `tail [fichier]`, permet d'afficher les dix dernières lignes d'un fichier dans la sortie du terminal. Par exemple : `tail .bashrc`, permet d'afficher les dix dernières lignes du fichier `.bashrc`, dans la sortie du terminal.

Afficher les 20 premières lignes du fichier ".bashrc"

La commande : `head [OPTION] [fichier]`, permet d'afficher les vingt premières lignes d'un fichier dans la sortie du terminal. Par exemple : `head -n 20 .bashrc`, permet d'afficher les vingt premières lignes du fichier `.bashrc`, dans la sortie du terminal.

Afficher les 20 dernières lignes du fichier ".bashrc"

La commande : `tail [OPTION] [fichier]`, permet d'afficher les vingt dernières lignes d'un fichier dans la sortie du terminal. Par exemple : `tail -n 20 .bashrc`, permet d'afficher les vingt dernières lignes du fichier `.bashrc`, dans la sortie du terminal.

Job 03

Fichier de documentation

Installer le paquet "cmatrix"

La commande : `sudo apt install nom_du_paquet`, permet d'installer un ou plusieurs paquets. Par exemple : `sudo apt install cmatrix`, permet d'installer le paquet `cmatrix`.

Lancer le paquet que vous venez d'installer

Pour lancer le paquet installé, il suffit d'écrire dans la console son nom.

Par exemple : `cmatrix`.

Mettre à jour son gestionnaire de paquets

La commande : `apt update`, permet de rechercher les mises à jour disponible du système et des programmes installés (sur les sources définies dans `/etc/apt/sources.list`).

```
yoann@debian:~$ su
Mot de passe :
root@debian:/home/yoann# apt update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Atteint :2 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
```

Mettre à jour ses différents logiciels

La commande : `apt upgrade`, permet d'installer les mises à jour identifiées avec `apt update`, sans supprimer les paquets installés. Contrairement à `apt full-upgrade`, qui peut supprimer des paquets si cela est nécessaire pour installer les nouvelles versions des paquets.

Télécharger les internets : Google

La commande : `wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb`, permet de télécharger le paquet Chrome. Une fois le paquet téléchargé, j'utilise la commande : `sudo dpkg -i google-chrome-stable_current_amd64.deb`, pour installer le navigateur à partir du package téléchargé.

Redémarrer votre machine

La commande : `sudo reboot`, permet de redémarrer la machine.

Éteindre votre machine

La commande : `sudo shutdown now`, permet d'éteindre la machine directement.

Job 04

Fichier de documentation

Créer un groupe appelé "Plateformeurs"

La commande : `sudo groupadd nom_groupe`, permet d'ajouter un groupe au système.

Par exemple : `sudo groupadd Plateformeurs`, création d'un groupe nommé Plateformeurs.

Créer un utilisateur appelé "User1"

La commande : `sudo useradd nom_utilisateur`, permet d'ajouter un utilisateur au système. Par exemple : `sudo useradd User1`, création d'un utilisateur nommé User1.

Créer un utilisateur appelé "User2"

La commande : `sudo useradd nom_utilisateur`, permet d'ajouter un utilisateur au système. Par exemple : `sudo useradd User2`, création d'un utilisateur nommé User2.

Ajouter "User2" au groupe Plateformeurs

La commande : `sudo adduser nom_utilisateur nom_groupe`, permet d'ajouter un utilisateur à un groupe.

Par exemple : `sudo adduser User2 Plateformeurs`.

```
yoann@debian:~$ sudo adduser User2 Plateformeurs
Ajout de l'utilisateur « User2 » au groupe « Plateformeurs »...
Adding user User2 to group Plateformeurs
Fait.
```

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

La commande : `cp [OPTION] SOURCE CIBLE`, permet de copier des fichiers à des répertoires. Par exemple : `cp users.txt droits.txt`, permet de copier le contenu du fichier `users.txt` dans `droits.txt`, si le fichier n'existe pas celui-ci va être créé automatiquement.

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

La commande : `cp [OPTION] SOURCE CIBLE`, permet de copier des fichiers à des répertoires. Par exemple : `cp users.txt groupes.txt`, permet de copier le contenu du fichier `users.txt` dans `groupes.txt`, si le fichier n'existe pas celui-ci va être créé automatiquement.

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

La commande : `chown nom_utilisateur nom_fichier`, permet de changer le propriétaire d'un fichier. Par exemple : `chown User1 droits.txt`, permet de mettre User1 propriétaire du fichier droits.txt.

```
yoann@debian:~/Job04$ ls -la
total 20
drwxr-xr-x  2 yoann yoann 4096  4 oct.  09:25 .
drwxr-xr-x 18 yoann yoann 4096  4 oct.  09:22 ..
-rw-r--r--  1 User1 yoann   12  4 oct.  09:24 droits.txt
-rw-r--r--  1 yoann yoann   12  4 oct.  09:25 groupes.txt
-rw-r--r--  1 yoann yoann   12  3 oct. 17:35 users.txt
```

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

La commande : `chmod [OPTION] MODE FICHIER`, permet de modifier les bits de comportements de fichiers. Par exemple, la commande `sudo chmod a+r droits.txt`, permet de donner la permission de lecture à tous les utilisateurs.

```
yoann@debian:~/Job04$ sudo chmod a+r droits.txt
```

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

La commande : `chmod [OPTION] MODE FICHIER`, permet de modifier les bits de comportements de fichiers. Par exemple, la commande `sudo chmod a+r groupes.txt`, permet de donner la permission de lecture à tous les utilisateurs pour le fichier groupes.txt.

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

La commande : `chmod [OPTION] MODE FICHIER`, permet de modifier les bits de comportements de fichiers. Par exemple, la commande `sudo chmod g+rw groupes.txt`, permet de donner la permission de lecture et d'écriture aux groupes pour le fichier groupes.txt.

```
yoann@debian:~/Job04$ sudo chmod g+rw groupes.txt
yoann@debian:~/Job04$ ls -la
total 20
drwxr-xr-x  2 yoann yoann 4096  4 oct.  09:25 .
drwxr-xr-x 18 yoann yoann 4096  4 oct.  09:22 ..
-rw-r--r--  1 User1 yoann   12  4 oct.  09:24 droits.txt
-rw-rw-r--  1 yoann yoann   12  4 oct.  09:25 groupes.txt
-rw-r--r--  1 yoann yoann   12  3 oct. 17:35 users.txt
```

Job 05

Fichier de documentation

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

La commande : `alias NAME="VALUE"`, permet de créer un alias, celui-ci sera supprimé après un reboot. Par exemple : `alias la="ls -la"`, permet de créer un alias nommé `la` qui exécutera la commande `ls -la`.

```
yoann@debian:~$ alias la="ls -la"
yoann@debian:~$ la
total 88
drwxr-xr-x 18 yoann yoann 4096 4 oct. 10:36 .
drwxr-xr-x  3 root  root  4096 3 oct. 11:24 ..
-rw-r--r--  1 yoann yoann 2177 4 oct. 10:08 .bash_history
-rw-r--r--  1 yoann yoann 220  3 oct. 11:24 .bash_logout
-rw-r--r--  1 yoann yoann 3526 3 oct. 11:24 .bashrc
drwxr-xr-x  2 yoann yoann 4096 3 oct. 11:25 Bureau
drwxr-xr-x 13 yoann yoann 4096 3 oct. 16:50 .cache
drwxr-xr-x 12 yoann yoann 4096 3 oct. 16:50 .config
drwxr-xr-x  2 yoann yoann 4096 3 oct. 11:25 Documents
drwxr-xr-x  2 yoann yoann 4096 3 oct. 15:54 .gnupg
drwxr-xr-x  2 yoann yoann 4096 3 oct. 11:25 Images
drwxr-xr-x  2 yoann yoann 4096 4 oct. 09:25 Job04
drwxr-xr-x  3 yoann yoann 4096 3 oct. 11:25 .local
drwxr-xr-x  2 yoann yoann 4096 3 oct. 11:25 Modèles
drwxr-xr-x  4 yoann yoann 4096 3 oct. 16:07 .mozilla
drwxr-xr-x  2 yoann yoann 4096 3 oct. 11:25 Musique
drwxr-xr-x  3 yoann yoann 4096 3 oct. 16:50 .pki
-rw-r--r--  1 yoann yoann 807  3 oct. 11:24 .profile
drwxr-xr-x  2 yoann yoann 4096 3 oct. 11:25 Public
drwxr-xr-x  2 yoann yoann 4096 3 oct. 11:25 .ssh
drwxr-xr-x  2 yoann yoann 4096 3 oct. 16:16 Téléchargements
```

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

La commande : `alias NAME="VALUE"`, permet de créer un alias, celui-ci sera supprimé après un reboot. Par exemple : `alias update="sudo apt-get update"`, permet de créer un alias nommé `update` qui exécutera la commande `sudo apt-get update`.

```
yoann@debian:~$ alias update="sudo apt-get update"
yoann@debian:~$ update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Atteint :2 http://deb.debian.org/debian bullseye-updates InRelease
Atteint :3 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :4 https://dl.google.com/linux/chrome/deb stable InRelease
Lecture des listes de paquets... Fait
```

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

La commande : `alias NAME="VALUE"`, permet de créer un alias, celui-ci sera supprimé après un reboot. Par exemple : `alias upgrade="sudo apt-get upgrade"`, permet de créer un alias nommé `upgrade` qui exécutera la commande `sudo apt-get upgrade`.

```
yoann@debian:~$ alias upgrade="sudo apt-get upgrade"
yoann@debian:~$ 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.
```


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

La commande : `USER=VALUE`, permet de créer une variable d'environnement nommé `USER`, qui sera égale à la valeur donnée. Par exemple, `USER=yoann`, donne la valeur `yoann`, à la variable `USER`.

Mettre à jour les modifications de votre `bashrc` dans votre shell actuel

La commande : `exec bash`, permet de recharger les modifications dans le fichier `.bashrc`, la commande remplace le shell `bash` actuel par une nouvelle instance.

Afficher les variables d'environnement

La commande : `printenv`, permet d'afficher les valeurs et les noms de toutes les variables d'environnement définies.

Ajouter à votre Path le chemin `"/home/votre utilisateur/Bureau"`

La commande : `export PATH=$PATH:/chemin/vers/le/répertoire`, permet d'ajouter un répertoire au `PATH`, celui-ci sera ajouté uniquement pour la durée de la session.

Par exemple, `export PATH=$PATH:/home/yoann/Bureau`, permet d'ajouter un répertoire/le chemin `/home/yoann/Bureau` au `PATH`.

```
yoann@debian:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
yoann@debian:~$ export PATH=$PATH:/home/yoann/Bureau
yoann@debian:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/yoann/Bureau
```

Job 06

Fichier de documentation

La commande : `tar [OPTION] fichier.tar.gz`, permet de décompresser un fichier `.tar.gz`. Par exemple, `tar -xf fichier.tar.gz x` pour extraire et `f` pour le fichier puis le nom du fichier à décompresser.

```
yoann@debian:~/Téléchargements$ tar -xf Ghost\ in\ the\ Shell.tar.gz
yoann@debian:~/Téléchargements$ ls
'Ghost in the Shell.pdf'  'Ghost in the Shell.tar.gz'
```

Job 07

Fichier de documentation

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

La commande : `echo 'texte' > fichier.txt`, permet de rediriger la sortie de `echo` et de créer des fichiers texte ainsi que d'écrire dans des fichiers textes existant ou non.

Opérateur de redirection `>`, le fichier est créé s'il n'existe pas. Si le fichier existe, la sortie `echo` est ajoutée au début du fichier, écrasant tout le contenu précédent.

Opérateur de redirection `>>`, le fichier est créé s'il n'existe pas. La sortie de `echo` est ajoutée à la fin du fichier n'écrasant pas le contenu du fichier existant.

Par exemple, `echo 'Je suis votre fichier texte' > une_commande.txt` permet de créer un fichier `une_commande.txt` contenant 'Je suis votre fichier texte'.

Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé “nb_lignes.txt”

La commande : `wc [OPTION] [FICHIER]`, permet d'afficher le nombre de lignes, de mots et d'octets du fichier voulu. Par exemple, `wc -l sources.list`, l'option `-l` permet d'afficher seulement le nombre de lignes de `sources.list`.

NB : faire une copie avant de le modifier afin de ne pas travailler dans le root, donc `cp /etc/apt/sources.list Job07/`, cette commande va copier puis créer un fichier `sources.list` dans le dossier `Job07`.

La commande : `wc -l sources.list > nb_lignes.txt`, permet de rediriger la sortie de `wc -l` dans un fichier `nb_lignes.txt`.

Première méthode :

```
yoann@debian:~/Job07$ wc -l sources.list > nb_lignes.txt
yoann@debian:~/Job07$ cat nb_lignes.txt
20 sources.list
yoann@debian:~/Job07$ ls
nb_lignes.txt  sources.list  une_commande.txt
```

Deuxième méthode:

```
yoann@debian:~/Job07$ wc -l sources.list > nb_lignes.txt && cat nb_lignes.txt
20 sources.list
yoann@debian:~/Job07$ ls
nb_lignes.txt  sources.list  une_commande.txt
```

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

La commande : `cat sources.list && cat sources.list > save_sources`, permet d'afficher dans un premier temps le contenu du fichier `sources.list` puis dans un deuxième temps avec les ET logique de faire une redirection de la sortie `cat` dans un fichier `save_sources`.

```
yoann@debian:~/Job07$ cat sources.list && cat sources.list > save_sources
# deb cdrom:[Debian GNU/Linux 11.5.0 _Bullseye_ - Official amd64 NETINST 2022091
0-10:38]/ bullseye main
```

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

La commande : `grep [OPTION] MOTIF`, permet d'afficher les lignes correspondant à un motif donné. Par exemple, `grep -R alias .`, cette commande permet de rechercher les fichiers `.` (par récursivité `-R`) contenant un `alias`.

```
yoann@debian:~$ grep -R alias .
grep: ./mozilla/firefox/lieltz02.default-esr/lock: Aucun fichier ou dossier de ce type
grep: ./mozilla/firefox/lieltz02.default-esr/storage/permanent/chrome/idb/3870112724rsegmnoittet-es.sqlite: binary file matches
grep: ./config/google-chrome/pnacl/0.57.44.2492/_platform_specific/x86_64/pnacl_public x86_64_pnacl sz nexex: binary file matches
grep: ./config/google-chrome/pnacl/0.57.44.2492/_platform_specific/x86_64/pnacl_public x86_64_pnacl llc nexex: binary file matches
grep: ./config/google-chrome/pnacl/0.57.44.2492/_platform_specific/x86_64/pnacl_public x86_64_ld nexex: binary file matches
./config/google-chrome/ZxcvbnData/1/passwords.txt:alias1
./config/google-chrome/ZxcvbnData/1/english_wikipedia.txt:aliasing
./config/google-chrome/ZxcvbnData/1/us_tv_and_film.txt:alias
./config/google-chrome/ZxcvbnData/1/us_tv_and_film.txt:aliases
grep: ./config/google-chrome/OnDeviceHeadSuggestModel/20220906.473124663/cr_fr_

./bash_history:alias man
./bash_history:man alias
./bash_history:alias la="ls -la"
./bash_history:alias upgrade="apt-get upgrade"
./bash_history:unalias upgrade
./bash_history:alias upgrade="sudo apt-get upgrade"
./bash_history:alias update="sudo apt-get update"
./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'
```

NB : je n'ai pas afficher la sortie entière du terminal mais seulement quelques lignes pour bien voir le résultat attendu.

Pour aller plus loin...

Fichier de documentation

Installer la commande tree

La commande : `sudo apt install nom_du_paquet`, permet d'installer un ou plusieurs paquets. Par exemple : `sudo apt install tree`, permet d'installer le paquet `tree`.

Lancer la commande tree en arrière-plan qui aura pour but d'afficher toute l'arborescence en enregistrant le résultat dans un fichier "tree.save"

Pour lancer le paquet installé, il suffit d'écrire dans la console son nom, `tree`.

La commande : `tree`, permet d'afficher l'arborescence.

La commande : `tree &` à gauche, permet de lancer `tree` en arrière plan.

L'opérateur : `'&'` permet de lancer deux commandes en simultanées dont la première commande sera lancée en arrière plan.

La commande : `tree & tree > Pour\ aller\ plus\ loin.../tree.save` à droite, permet de lancer `tree` en arrière plan ainsi que de l'enregistrer dans le fichier `tree.save` situé dans le dossier `Pour\ aller\ plus\ loin.../`.

```
yoann@debian:~$ tree &
[1] 3355
yoann@debian:~$ .
├── Bureau
├── Documents
├── Images
├── Job04
│   ├── droits.txt
│   ├── groupes.txt
│   └── users.txt
├── Job07
│   ├── nb_lignes.txt
│   ├── save_sources
│   ├── sources.list
│   └── une_commande.txt
├── Modèles
├── Musique
├── Pour aller plus loin...
├── Public
├── Téléchargements
│   ├── Ghost in the Shell.pdf
│   └── Ghost in the Shell.tar.gz
└── Vidéos

11 directories, 9 files
[1]+  Fini          tree

yoann@debian:~$ tree & tree > Pour\ aller\ plus\ loin.../tree.save
[1] 3359
yoann@debian:~$ .
├── Bureau
├── Documents
├── Images
├── Job04
│   ├── droits.txt
│   ├── groupes.txt
│   └── users.txt
├── Job07
│   ├── nb_lignes.txt
│   ├── save_sources
│   ├── sources.list
│   └── une_commande.txt
├── Modèles
├── Musique
├── Pour aller plus loin...
│   └── tree.save
├── Public
├── Téléchargements
│   ├── Ghost in the Shell.pdf
│   └── Ghost in the Shell.tar.gz
└── Vidéos

11 directories, 10 files
[1]+  Fini          tree
```

Lister les éléments présents dans le dossier courant est utilisé directement le résultat de votre première commande pour compter le nombre d'éléments trouvés

La commande : `pwd`, permet d'afficher le répertoire courant.

La commande : `pwd | wc -l`, permet d'envoyer la sortie standard de `pwd` vers l'entrée standard de `wc -l`.

L'opérateur : `'|'` permet d'envoyer la sortie standard d'un programme vers l'entrée standard d'un second programme.

Voici deux façons de le faire, la première en une ligne et la deuxième en deux lignes.

La première méthode est possible grâce au ET logique, donc `&&`.

L'opérateur : '`&&`' permet de lancer une commande si et seulement si la première commande s'est correctement terminée.

```
yoann@debian:~$ pwd && pwd | wc -l
/home/yoann
1
yoann@debian:~$ pwd
/home/yoann
yoann@debian:~$ pwd | wc -l
1
```

Lancer une commande pour update vos paquets, si l'update réussit alors, vous devrez lancer un upgrade de vos paquets. Si l'update échoue, votre upgrade ne se lancera pas

La commande : `apt update`, permet de rechercher les mises à jour disponible du système et des programmes installés.

La commande : `apt upgrade`, permet d'installer les mises à jour identifiées avec `apt update`, sans supprimer les paquets installés.

Pour réaliser l'opération voulue dans la consigne il faut utiliser l'opérateur '`&&`', qui permet de lancer une commande si et seulement si la première commande s'est correctement terminée.

La commande : `sudo apt update && sudo apt upgrade`, nous permet de réaliser cette opération.

```
yoann@debian:~$ sudo apt update && sudo apt upgrade
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Atteint :4 https://dl.google.com/linux/chrome/deb stable InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
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.
```

Bonus :)

Fichier de documentation

Installer SSH

La commande : `apt-cache policy openssh-server`, permet de vérifier si OpenSSH est bien installé sur notre serveur Linux. Comme ce n'est pas le cas, je vais devoir l'installer avec la commande `sudo apt-get update && sudo apt-get install openssh-server && apt-get install ssh`.

```
yoann@debian:~$ apt-cache policy openssh-server
openssh-server:
  Installé : (aucun)
  Candidat : 1:8.4p1-5+deb11u1
  Table de version :
    1:8.4p1-5+deb11u1 500
      500 http://deb.debian.org/debian bullseye/main amd64 Packages
yoann@debian:~$ sudo apt-get update && sudo apt-get install openssh-server
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Atteint :4 https://dl.google.com/linux/chrome/deb stable 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
Les paquets supplémentaires suivants seront installés :
  openssh-sftp-server runit-helper
Paquets suggérés :
  molly-guard monkeysphere ssh-askpass ufw
Les NOUVEAUX paquets suivants seront installés :
  openssh-server openssh-sftp-server runit-helper
0 mis à jour, 3 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 446 ko dans les archives.
Après cette opération, 1 765 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] y
```

NB : je n'ai pas afficher la sortie entière du terminal mais seulement quelques lignes pour bien voir le résultat attendu.

Générer une clé SSH

La commande : `ssh-keygen`, permet de générer une clé SSH.

```
yoann@debian:~/Bonus$ ssh-keygen && ls
Generating public/private rsa key pair.
Enter file in which to save the key (/home/yoann/.ssh/id_rsa): SSH_KEY
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in SSH_KEY
Your public key has been saved in SSH_KEY.pub
The key fingerprint is:
SHA256: [redacted]@debian
The key's randomart image is:
+---[RSA 3072]-----+
|  o . * . .      |
| + % . . +      |
| o %oE. .      |
| ..@ =o        |
| O+..S.        |
| *.oo+.        |
| =.B.          |
| +o*.          |
| **.          |
+----[SHA256]-----+
SSH_KEY  SSH_KEY.pub
```

Se connecter à une VM ou l'ordinateur d'un camarade via SSH

La commande : `ssh nom_utilisateur@adresse_ip`, permet de se connecter au serveur d'un camarade.

```
yoann@debian:~$ ssh root@[redacted]
The authenticity of host '[redacted]' can't be established.
ECDSA key fingerprint is SHA256:n/xnhMx5ufBm28EWuMa/PByNeZk55Q0V3CwWCZMv/Y8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[redacted]' (ECDSA) to the list of known hosts.
root@[redacted] password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 3.10.0-1160.53.1.vz7.185.3 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Tue Oct  4 16:02:52 2022 from 37.26.187.6
root@alex:~# exit
logout
Connection to [redacted] closed.
```

Configurer SSH pour empêcher le login root (root ne peut pas se connecter en SSH)

La commande : `sudo nano /etc/ssh/sshd_config`, permet d'éditer le fichier `/sshd_config`. Il faut décommenter la ligne `PermitRootLogin`, puis la lui mettre `no` comme paramètre afin d'empêcher le root de se connecter.

```
GNU nano 6.2 /etc/ssh/sshd_config
# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Modifier le port de connexion de SSH (autre que 22)

La commande : `sudo nano /etc/ssh/sshd_config`, permet d'éditer le fichier `/sshd_config`. Il suffit de décommenter la ligne du `Port` et de renseigner un autre chiffre pour lui en attribuer un nouveau.

```
GNU nano 6.2 /etc/ssh/sshd_config *
Include /etc/ssh/sshd_config.d/*.conf

Port 007
#AddressFamily any
ListenAddress 0.0.0.0
#ListenAddress ::
```

Ensuite se connecter en SSH sans avoir à renseigner de mot de passe

La commande : `sudo nano /etc/ssh/sshd_config`, permet d'éditer le fichier `/sshd_config`. Dans le fichier `/sshd_config`, il suffit de décommenter la ligne `PasswordAuthentication` et de mettre cette ligne en non. On pourra donc se connecter en SSH sans avoir à renseigner le mot de passe.

```
GNU nano 6.2 /etc/ssh/sshd_config *
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```


Uploader un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade)

La commande : `scp /home/yoann/fichier.txt root@adresse_ip:/répertoire`, permet d'envoyer un fichier comme `fichier.txt` de mon ordinateur à un utilisateur distant, possédant une `adresse_ip`, dans un `/répertoire` connu à l'avance.

Télécharger un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade)

La commande : `scp root@adresse_ip:/répertoire/fichier.txt /home/yoann`, permet de télécharger un fichier comme `fichier.txt` situé dans le `/répertoire` d'un utilisateur distant, celui-ci doit être connu à l'avance. De plus son `adresse_ip` doit être renseignée afin de pouvoir accéder à ses fichiers.

Questions

Quel est l'intérêt d'utiliser SSH ?

SSH : Secure Shell ou Socket Secure Shell.

Le protocole SSH est un protocole réseau qui offre aux utilisateurs, en particulier aux administrateurs système, un moyen sécurisé d'accéder à un ordinateur via un réseau sécurisé.

Est-ce que les clés générées par SSH par défaut sont assez sécurisées ? Justifier votre réponse

Secure Shell fournit une authentification forte par mot de passe et une authentification par clé publique, ainsi que des communications de données cryptées entre deux ordinateurs se connectant sur un réseau ouvert, tel qu'internet.

Citez d'autres protocoles de transfert ? Quelles sont les différences entre ses protocoles ?

Plusieurs autres protocoles de transfert existent comme Telnet, HTTP, FTP, SMTP/IMAP, SSL/TLS.

SSH & Telnet : Telnet est l'un des premier protocoles d'application internet, l'autre est FTP. SSH et Telnet sont similaires sur leur fonctionnement mais diffèrent sur un point, le protocole SSH utilise la cryptographie à clé publique pour authentifier les points de terminaison lors de la configuration d'une session de terminal, ainsi que pour chiffrer les commandes et la sortie de la session. Telnet est principalement utilisé pour l'émulation de terminal, alors que SSH peut être utilisé pour faire de l'émulation de terminal, ainsi que pour émettre des commandes à distance.

SSH & SSL/TLS : le protocole Transport Layer Security met à jour le protocole Secure Socket Layer. Conçu pour assurer la sécurité des transmissions réseau au niveau de la couche transport. Le protocole SSH fonctionne également au niveau de cette couche. Sous TLS seul le serveur est authentifié avec une paire de clés, or SSH utilise une paire de clés distincte pour authentifier chaque connexion. De plus, TLS permet aux connexions d'être chiffrées sans authentification ou authentifiées sans chiffrement, or SSH crypte et authentifie toutes les connexions.