**JOB 1:**

**- Afficher le manuel de la commande ls**

> ls --help ou > man ls

**- Afficher les fichiers cachés du home de votre utilisateur**

> ls -a

“-a” permet d’afficher les entrées commençant par “.”, donc les fichiers cachés.

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

> ls -a -l

“-l” permet de lister les fichiers en affichant des données complémentaires, comme les droits d’accès au fichiers par exemple.

**- Comment ajouter des options à une commande ?**

On ajoute des options à une commande en les déclarant après le nom de la commande. Ces options sont précédées d’un tiret “-”.

**- Quelles sont les deux syntaxes principales d’écriture des options pour une**

**commande ?**

On peut choisir de mettre les arguments en “série” (ls **-a -l**) ou “collés” (ls **-al**).

**JOB 2:**

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

> cat .bashrc

La commande “cat” (concaténer) permet récupérer le contenu d’un fichier et l’écrire dans l’affichage de sortie spécifié (console par défaut).

**- afficher les 10 premières lignes du fichier “.bashrc”**

> head -10 .bashrc

La commande “head” récupère les n premières lignes d’un fichier et l’écrit dans l’affichage de sortie spécifié (console par défaut).

**- afficher les 10 dernières lignes du fichier “.bashrc”**

> tail -10 .bashrc

“tail” récupère les n dernières lignes d’un fichier et l’écrit dans l’affichage de sortie spécifié (console par défaut).

**- afficher les 20 premières lignes du fichier “.bashrc”**

> head -20 .bashrc

**- afficher les 20 dernières lignes du fichier “.bashrc”**

> tail -20 .bashrc

**JOB 3:**

**- Installer le paquet “cmatrix”**

> sudo apt-get install cmatrix

**- lancer le paquet que vous venez d’installer**

> cmatrix

**- Mettre à jour son gestionnaire de paquets**

> sudo apt-get --only-upgrade install cmatrix

“--only\_upgrade” permet d’installer seulement les mises à jour du paquet cmatrix.

**- Mettre à jour ses différents logiciels**

> sudo apt update

Cette commande va comparer les versions des paquets installés et les mettre à jour s’il ne l’est pas.

**- Télécharger les internets : Google**

> wget https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb

Permet de récupérer le fichier spécifié dans l’URL. (Web Get)

> sudo dpkg -i google-chrome-stable\_current\_amd64.deb

dpkg (Debian Package) est un gestionnaire de paquet permettant notamment d’installer des paquets .deb.

**- Redémarrer votre machine**

> sudo reboot ou > sudo shutdown -r now

Commandes similaires, mais “shutdown” permet de spécifier à quel moment on souhaite effectuer un redémarrage (ici “now” pour “maintenant”).

**- éteindre votre machine**

> sudo shutdown -P now

“-P” symbolise “power off”

**JOB 4:**

**- Créer un fichier users.txt qui contiendra User1 et User2 séparé par un retour à la ligne**

> nano users.txt

La commande nano affiche un éditeur de texte dans le terminal.

**- Créer un groupe appelé “Plateformeurs”**

> sudo groupadd Plateformeurs

groupadd est une commande permettant d’ajouter un groupe utilisateur.

**- Créer un utilisateur appelé “User1”**

> sudo useradd User1

useradd est une commande permettant de créer un utilisateur.

**- Créer un utilisateur appelé “User2”**

> sudo useradd User2

**- Ajouter “User2” au groupe Plateformeurs**

> sudo usermod -a -G Plateformeurs User2

usermod permet de gérer des paramètres relatifs à un utilisateur, comme le groupe auquel il appartient ou son mot de passe.

**- Copier votre “users.txt” dans un fichier “droits.txt”**

> cat users.txt > droits.txt

Concatène le contenu de “users.txt” dans “droits.txt”.

**- Copier votre “users.txt” dans un fichier “groupes.txt”**

> cat users.txt > groupes.txt

Concatène le contenu de “users.txt” dans “groupes.txt”.

**- Changer le propriétaire du fichier “droits.txt” pour mettre “User1”**

> sudo chown User1 droits.txt

chown permet de changer le propriétaire du fichier cible.

> ls -l droits.txt (pour vérifier l’ownership)

**- Changer les droits du fichier “droits.txt” pour que “User2” ai accès seulement en**

**lecture**

> sudo chmod o=r groupes.txt

chmod permet de modifier les droits d’accès du fichier cible.

“o=r” signifie que pour les utilisateurs qui ne sont ni les propriétaires du fichier ni dans le groupe spécifié du fichier, on affecte seulement l’accès en lecture (symbolisé par le r).

<https://www.guru99.com/file-permissions.html>

**- Changer les droits du fichier “groupes.txt” pour que les utilisateurs puissent**

**accéder au fichier en lecture uniquement**

> sudo chmod u=r groupes.txt

Change les droits de telle sorte que les utilisateurs aient seulement accès au fichier en écriture.

**- Changer les droits du fichier pour que le groupe “Plateformeurs” puissent y**

**accéder en lecture/écriture.**

> sudo chgrp -c Plateformeurs groupes.txt

chgrp permet de changer l’appartenance du groupe du fichier spécifié.

> sudo chmod g=rw groupes.txt

Change les accès du groupe propriétaire du fichier en lecture/écriture (rw).

<https://www.guru99.com/file-permissions.html>

**JOB 5:**

**- Ajouter un alias qui permettra de lancer la commande “ls -la” en tapant “la”**

> alias la=”ls -la”

Créer un alias “la” permettant d’exécuter la commande “ls -la”.

**- Ajouter un alias qui permettra de lancer la commande “apt-get update” en tapant**

**“update”**

> alias update=”apt-get update”

Créer un alias “update” permettant d’exécuter la commande “apt-get update”.

**- Ajouter un alias qui permettra de lancer la commande “apt-get upgrade” en**

**tapant “upgrade”**

> alias upgrade=”apt-get upgrade”

Créer un alias “upgrade” permettant d’exécuter la commande “apt-get upgrade”.

**- Ajouter une variable d’environnement qui se nommera “USER” et qui sera égale à**

**votre nom d’utilisateur**

> export USER="erwan"

Ajoute la variable USER égale au nom d’utilisateur dans les variables d’environnement.

**- Mettre à jour les modifications de votre bashrc dans votre shell actuel**

> source ~/.bashrc

“source” permet de définir pour le shell actuel certains paramètres, dont des alias par exemple.

**- Afficher les variables d’environnement**

> export

**- Ajouter à votre Path le chemin "/home/'votre utilisateur'/Bureau"**

> export PATH=$PATH:/home/erwan/Bureau

Ajoute l’entrée “/home/erwan/Bureau” à la variable déjà existante “PATH”.

**JOB 6:**

**Vous devez télécharger l’archive suivante et la désarchiver seulement avec le terminal.**

> tar -xf "Ghost in the Shell.tar.gz"

La commande tar permet de gérer, créer et extraire des archives.

**JOB 7:**

UTILISER 1 SEULE COMMANDE

* **Créer un fichier “une\_commande.txt” avec le texte suivant “Je suis votre fichier texte”**

> echo "Je suis votre fichier texte" > une\_commande.txt

Redirige la sortie de la commande echo vers un fichier texte. “>” écrase le contenu du fichier et “>>” ajoute le texte au reste du fichier.

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

> less /etc/apt/sources.list | wc -l > nb\_lignes.txt

La commande “wc -l” qui permet de compter le nombre de ligne du texte qu’on lui

donne, va prendre ici comme texte la sortie de la commande “less

/etc/apt/sources.list”, qui renvoie le contenu du fichier source.list. La sortie de “wc -l”

est ensuite redirigée dans le fichier nb\_lignes.txt pour sauvegarder le nombre de

lignes comptées.

* **Afficher le contenu du fichier source apt et l’enregistrer dans un autre fichier appelé “save\_sources”**

> less /etc/apt/sources.list | tee save\_sources

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

> grep --include=".\*" -rnw . -e $(less MOT\_A\_CHERCHER.file)

“grep” est une commande permettant de chercher une chaîne de caractère dans des fichiers. L’argument --include=”.\*” permet de chercher seulement dans des fichiers commençant par “.” (les fichiers cachés), où “\*” spécifie que l’on cherche des fichiers se terminant par n’importe quelle expression.

“-rnw .” spécifie le dossier (ici le dossier courant spécifié avec “.”) et la manière dont on cherche ces fichiers (r indiquant de manière récursive, donc tous les sous-dossiers, n permet d’afficher la ligne où le mot a été repéré, et w spécifie qu’on recherche des mots entiers).

“-e” spécifie l’expression que l’on souhaite rechercher.

Cette expression est récupérée depuis un fichier grâce à la commande “less” ($(less MOT\_A\_CHERCHER.file)).

**Pour aller plus loin :**

UTILISER 1 SEULE COMMANDE

* **Installer la commande tree**

> sudo apt-get install tree

* **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”**

> tree | tee tree.save

“tee” permet de diriger la sortie de la commande avant la “barre verticale” à la fois dans l’affichage standard et dans une autre “sortie”, qui ici est un fichier.

* **lister les éléments présents dans le dossier courant et utiliser directement le résultat de votre première commande pour compter le nombre d’éléments trouvés**

> ls | wc -l

Compte le nombre d’éléments du dossier courant en donnant comme argument à la commande “wc -l” le nombre de lignes que renvoie la commande “ls”.

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

> sudo apt update && sudo apt upgrade

L’opérateur “&&” permet de lancer la commande suivante si la précédente réussit.

À l’inverse, l’opérateur “||” permet de lancer la commande suivante si la précédente échoue.

**BONUS :**

**Installer SSH**

> sudo apt-get install ssh

**Générer une clé SSH**

> ssh-keygen

Génère une clé SSH de 2048 bits par défaut.

Pour spécifier un type de clé différent, on peut utiliser la commande:

> ssh-keygen -t rsa -b 4096

où “-t” spécifie le type de clé (RSA ou DSA) et “-b” sa taille (en bits).

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

> ssh -p 22 erwan@<ip ou nom de domaine>

“-p” pour spécifier le port, et “erwan@<ip ou nom de domaine>” pour l’hôte.

> ssh “erwan blancard@<ip ou nom de domaine>”

Connection de la VM à l’hôte (NAT).

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

> sudo nano /etc/ssh/sshd\_config

Ajouter la ligne “PermitRootLogin no”

**Modifier le port de connexion de SSH (autre que 22) pour la prochaine connexion:**

> ssh -p <port> erwan@<ip ou nom de domaine>

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

> ssh-copy-id -i ~/.ssh/id\_rsa.pub erwan@<ip ou nom de domaine>

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

> scp <chemin fichier source> user@domain:<destination>

Pour spécifier une source qui provient du PC distant, on place “user@domain:”avant la source.

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

> scp user@domain:<chemin fichier source> <destination>

Pour spécifier une source qui provient du PC distant, on place “user@domain:”avant la source.

**Limiter l’utilisation de SSH à un groupe particulier nommé “Plateforme\_ssh”**

Dans le dossier de configuration sshd\_config, ajouter la ligne “AllowGroups Plateforme\_ssh”.

* **Quel est l’intérêt d’utiliser SSH ?**

SSH permet d’accéder à un shell sur une machine distant depuis son ordinateur. Cet accès est chiffré par une clé que possèdent les 2 machines.

Cela permet également d’accéder plus facilement au shell sur des machines ne possédant pas d’interface graphique, comme des serveurs ou des appareils embarqués par exemple.

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

Les clés générées par défaut par SSH sont des clés RSA d’une taille de 2048 bits. Pour comparaison, les sites utilisant la protection SSL (https) ont une clé ne pouvant dépasser les 256 bits.

Une clé par défaut générée par SSH est donc 8 fois plus grande que celles utilisées par SSL, et donc plus sécurisée.

Par ailleurs, lorsque l’on génère une clé, on se retrouve avec 2 clés: une clé publique et une clé privée, et pour déchiffrer des messages envoyés, on a besoin des 2 clés. C’est le principe des clés asymétriques: une clé ne sert à rien sans l’autre.

On peut ainsi partager sa clé publique avec quelqu’un pour qu’il puisse envoyer des messages chiffrés avec cette clé pour ensuite les déchiffrer avec notre clé privée.

* **Citez d’autres protocoles de transfert ? Quelles sont les différences entre ces protocoles ?**
* Telnet : S’appuie sur le protocole TCP/IP. Utilise le port 23 par défaut. Protocole utilisé pour communiquer avec des serveurs principalement, mais les données envoyées sur le réseau ne sont pas chiffrées.
* rlogin : Utilise le port 513 par défaut. Permet de se connecter à distance sur des machines Unix. Le principal défaut de sécurité est que ce système repose sur une “machine hôte de confiance”, qui permet à cette machine de ne pas fournir obligatoirement de mot de passe lors de connexions. Ce problème est transitif car si une machine A fait confiance à B, et que B fait confiance à C, C aura accès à A sans faire explicitement confiance à C.
* rcp : (remote procedure call) Permet de faire appel à des procédures sur un ordinateur distant grâce à une “application serveur“ qui tourne sur celle-ci. Utilise le système d’authentification “Diffie-Hellman”.