
RunTrack1: SHELL

Sujet 1 / 2 : Paramètres et lectures

Sujet 3 : Installation de logiciels en CLI

Sujet 4 : Permissions de fichiers

Sujet 5 : Conforts en CLI

Sujet 6 / 7 : Redirections, Tubes

PELAYO Joël

Sujet 1 / 2 : Paramètres et lectures

- Commande ls : affiche le contenu du répertoire actuel.
 - Commandes de la consigne :
 - **man ls**
 - **ls -a**
 - **ls -al**
 - **ls 'nom de répertoire/chemin'** : listera les éléments contenus dans le répertoire/chemin spécifié.
 - **ls *** : liste tous les éléments dans le répertoire actuel et les répertoires présents.
 - Exemple d'options :
 - **ls -d** : liste les répertoire sans leur contenu.
 - **ls -R** : liste de manière récursive dans tous les répertoires à partir de la position où l'on se trouve.
 - **ls -a** : liste tous les éléments présents en plus d'éléments cachés (souvent pour protéger des fichiers systèmes essentiels par exemple).
 - **ls -l** : liste les éléments et leurs droits d'accès.
 - On peut également combiner les différentes options :
ls -aR

Les commandes en CLI peuvent prendre le plus souvent ce qu'on appelle des paramètres (ou arguments ou autres encore) qui servent d'options pour la commande spécifié, changeant donc son comportement ou sa sortie.

Il faut savoir qu'un espace pour le SHELL sépare les paramètres, fichiers ou autres, c'est pour cela qu'on devra éviter de nommer des fichiers avec des espaces afin d'éviter toute confusion :

- '**commande**' '**paramètre1**' (dans cet exemple, paramètre1 peut prendre plusieurs options comme on peut le voir au-dessus '**ls -aR**')
 - '**commande**' '**paramètre1**' '**paramètre2**'

Il est possible qu'une commande peut prendre plus de 1 paramètres et seront donc différencier avec un autre espace et/ou une autre syntaxe (principalement '-' pour le premier, '--' pour le deuxième ou sert aussi de syntaxe différente pour un même paramètre):

- Commande cat : affiche le contenu d'un fichier texte, il permet aussi d'en créer (cat > 'fichier')
 - Commande de la consigne : **cat .bashrc** (echo fonctionne aussi)
- Commande head : affiche le contenu d'un fichier texte à partir du haut, on peut spécifier le nombre de ligne en paramètre -n, n étant un nombre. (head -n10 'fichier' : affiche les 10 premières lignes)
 - Commande de la consigne :
 - **head -n10 .bashrc**
 - **head -n20 .bashrc**
- Commande tail : commande head mais depuis le bas.
 - Commande de la consigne :
 - **tail -n10 .bashrc**
 - **tail -n20 .bashrc**

Sujet 3 : Installation de logiciels en CLI

Sous Windows on est habitué à télécharger un logiciel en prenant son installateur, lancer l'installateur et enfin obtenir le logiciel. Sur Linux il faudra pour la plupart des logiciels passer via un gestionnaire de paquets (graphique ou en CLI). Un paquet est comme son nom l'indique, un paquet de fichiers liés pouvant être installés/supprimé en bloc via le gestionnaire de paquets. Le gestionnaire de paquets permet donc de gérer ces paquets, on pourra installer/supprimés/mettre à jour nos paquets (du coup logiciel).

Un gestionnaire de paquets peut posséder un dépôt de paquets, il s'agit d'un serveur contenant différents paquets qui seront directement disponible, cela permet d'éviter d'aller chercher un paquet sur internet, s'il est dans le dépôt il suffira de passer par une commande du gestionnaire de paquets.

Sous Ubuntu (Debian) :

- dpkg : Gestionnaire de paquets extérieur au dépôts apt.
- apt (ou apt-get) : Gestionnaire de paquets Ubuntu

Sous Arch Linux :

- pacman : Gestionnaire de paquets Arch
- AUR : Dépôt communautaire de Arch, les utilisateurs peuvent y déposer des paquets qu'ils ont créé par exemple

- Commandes de la consigne :
 - **sudo apt install cmatrix**
 - **cmatrix**
 - **sudo apt update**
 - **sudo apt upgrade**
 - Télécharger google chrome :
 - **wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb**
 - **sudo dpkg google-chrome-stable_current_amd64.deb**
 - **reboot**
 - **lol**

Sujet 4 : Utilisateurs/Permissions de fichiers

- Commandes de la consigne :
 - **groupadd Plateformeurs**
 - **adduser user1**
 - **adduser user2**
 - **usermod -aG user2 Plateformeurs**
 - **cp users.txt droits.txt**
 - **cp users.txt groupes.txt**
 - **chown user1 droits.txt**
 - **chmod o-wx droits.txt**
 - **chmod u-wx groupes.txt**
 - **chmod g-x groupes.txt**

Les principales commandes à retenir sont :

- chown
- chmod
 - setfacl
 - getfacl
- groupadd
- adduser

Ces commandes permettent de gérer l'accès/modification des fichiers et donc éventuellement de gérer la sécurité sur son système.

Sujet 5 : Conforts en CLI

Il est possible de configurer des variables ou des raccourcis sur un terminal via les fichiers de configuration de son terminal.

- Commandes de la consigne :
 - **alias la='ls -al'**
 - **alias update='apt-get update'**
 - **alias upgrade='apt-get upgrade'**
 - **export USER=\$(whoami)**
 - **source .bashrc**
 - **printenv**
 - **export PATH=/home/'utilisateur'/Bureau**

Sujet 6 / 7 : Redirections, tubes

Sujet 6 (Commandes):

- **tar -xzf 'Ghost In The Shell.tar.gz'**
- **xdg-open 'Ghost In The Shell.pdf'**

Sujet 7 / pour aller plus loin :

L'utilisation de redirections et de tubes permet de d'enchaîner des commandes à la suite ou entre elles.

- **<, >, <<, >>** : Permet de rediriger l'entrée ou la sortie d'une commande vers une autre
- **|** : Permet d'enchaîner les commandes, on peut utiliser une commande sur la sortie d'une commande. Elles peuvent d'ailleurs s'enchaîner.
- **&, &&** : Permet de lancer simultanément (&) ou à la suite (&&) différentes commandes à la suite en une ligne
- Commande de la consigne :
 - **echo 'Je suis votre fichier texte' > une_commande.txt & cat /etc/apt | wc -l > /home/'utilisateur'/Documents/nb_lignes.txt & grep -r alias .***
- Pour aller plus loin :
 - **sudo apt install tree && tree / > /home/'utilisateur'/Documents/tree.save && ls | wc -w & sudo apt update && sudo apt upgrade**

Bonus :

Le ssh (Secure Shell) est un protocole de communication TCP/IP sécurisé de la couche application du modèle OSI orienté vers des sessions à distance.

- Commandes du bonus :
 - Installer et démarrer ssh :
 - **sudo apt install ssh && sudo apt install openssh-server && sudo apt install openssh-client**
 - **sudo systemctl enable ssh**
 - **sudo systemctl start ssh**
 - **ssh-keygen**
 - **ssh <id>@<ipaddress> [-p ['port']]**
 - Connection sans mdp :
 - Obtenir la clé public ssh d'un hôte et la mettre dans ~/.ssh
 - **sudo nano /etc/ssh/sshd_config** et modifier/décommenter **'#PasswordAuthentication no'**
 - Connection via la commande précédente
 - Modifier le root login/port :
 - **sudo nano /etc/ssh/sshd_config**
 - Pour le root login, modifier et décommenter **'#PermitRootLogin no'**
 - Pour le port, modifier et décommenter **'#Port <numéro de port>'**
 - **sudo systemctl restart ssh**
 - Upload/Download :
 - **scp 'fichier à uploader' <ip>@<ipaddress> : 'path'**
 - **scp <ip>@<ipaddress> : '/'pathtofile' 'path'**
 - Limiter l'utilisation à un groupe :
 - **groupadd Plateforme_ssh**
 - **sudo chown :Plateforme_ssh /bin/ssh**
 - **sudo chmod o-rwx /bin/ssh**
 - **usermod -aG Plateforme_ssh 'utilisateur'**

L'intérêt du protocole ssh permet d'avoir un moyen de connexion à distance sécurisé via une clé de chiffrement.

Le chiffrement des clés ssh utilisent 3 mécanismes : la symétrie et l'asymétrie (donc un chiffrement hybride), et le hashing. Via ces mécanismes, les clés ssh devraient être assez sécurisés.

- Symétrie : Se sert d'une clé de chiffrement publique qui doit être partagé au correspondant de manière confidentielle, tout ceux qui ont la clé peuvent déchiffrer le fichier 'protégée'
 - Asymétrie : Se sert d'une clé de chiffrement (publique) et d'une clé de déchiffrement (privée), elles peuvent être utilisé de différente manière mais généralement la clé privée ne doit pas être divulguée, et la clé publique est partagé avec ceux avec qui on souhaite partager un fichier.
 - Hashing : Il s'agit de la manière dont on chiffre les clés ou les mots de passe. La clé ou le mdp passe est 'hasher' par une fonction de hachage pour donner une donnée indéchiffrable qui sera reliée à la clé ou mdp, le tout étant structuré par une base de données structuré de tel manière à relié la donnée indéchiffrable avec la clé ou mdp.
- Différents protocoles de transfert :
 - FTP/FTPS (File Transfer Protocol) : Protocole de partage de fichier entre appareils sur un réseau TCP/IP. FTPS est sécurisé par TLS qui est un protocole de sécurisation.
 - SFTP : FTP mais sécurisé par SSH.
 - SCP (Secure Copy Protocol) : Protocol de transfert de données en s'appuyant sur SSH.
 - HTTP/HTTPS (Hypertext transfer protocol): Protocole de transfert pour les fichiers multimédia, est particulièrement utilisé pour le web et ses serveurs et donc notamment des fichiers HTML.

La différence entre ces différents protocoles de transfert de la couche application se trouve dans leur méthode de transfert, le type de données, ou leur méthode pour sécuriser les données qu'ils transferts. Certains auront des processus plus adaptés que d'autres.