



TP 1 : Découverte du Shell

Objectifs :

- Développer la logique, la compréhension du système Linux et l'autonomie dans l'utilisation du terminal.
- Se familiariser avec la navigation dans le système Linux grâce à l'utilisation des commandes de base du Shell.

Modalités pédagogiques :

Travail individuel

Durée : 1 jour

Pré-requis :

Poste sous Ubuntu ou Debian

Consignes :

Pour chaque partie du TP, vous devez remettre un document avec toutes les commandes utilisées. Des capture d'écran suffisant 😊

Un peu de théorie

Prenez le temps de lire ce cours, on débriefera dans 15min 😎 :

https://www.canva.com/design/DAGIRwVTGe4/KBybCDf8ucO1es5h1CbXxw/edit?utm_content=DAGIRwVTGe4&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Passe à la pratique

1. Découvrir les premières commandes du Shell Linux :

Réaliser les tâches suivantes en ligne de commandes :

- Changer le mot de passe de votre session, fermer puis rouvrir la session pour vérifier la prise en compte du mot de passe.
- Quel utilisateur est connecté ? Depuis quelle heure ? Quel est le temps d'activité de cet utilisateur ? Quel est le numéro de processus de votre session ?

- À quel groupe(s) appartenez-vous ? Quels-sont vos identifiants d'utilisateur (UID) et de groupes (GID) ?
- Afficher la date et l'heure.

2. Se déplacer dans des répertoires et manipuler des répertoires et des fichiers :

2.1. Localiser son répertoire courant et afficher son contenu

- Qu'est-ce qu'un répertoire courant ? Afficher le répertoire courant, quelle-est la spécificité de ce répertoire ?
- Existe-t-il des fichiers et/ou des répertoires dans votre répertoire courant ? Le(s)quel(s) ?

2.2. Création de répertoire et de fichier

- Créer un répertoire nommé MonRepertoire puis se placer dans ce répertoire
- Créer un fichier nommé MonPremierFichier

2.3. Afficher les informations détaillées des répertoires et des fichiers

- Afficher les droits attribués concernant le répertoire et le fichier que vous venez de créer

2.4. Rechercher des fichiers particuliers

- Afficher les fichiers se terminant par « .conf » dans le répertoire /etc

2.5. Créer une arborescence de répertoires

- Dans votre répertoire de travail (/home/votre_login), créer l'arborescence suivante :

```
/home
  votre_login
    TP
    DOCUMENTS
    PERSO
    PHOTOS
    DIVERS
    ARCHIVES
    EXECUTABLES
```

- Créer un fichier dans le répertoire photos
- Supprimer le répertoire TP que vous venez de créer
- Supprimer le répertoire PERSO que vous venez de créer

3. Manipuler le contenu des fichiers

- Créer un fichier appelé MonDeuxiemeFichier dans votre home à l'aide de la commande **touch**
- Quels-sont les éditeurs de texte en CLI principaux de Linux ?
- Modifier le fichier précédemment créé avec l'éditeur de texte de votre choix avec le contenu suivant :



LINUX, c'est trop génial, mais les interfaces graphiques me manquent quand même...

- Créer un autre fichier appelé MonTroisiemeFichier dans votre home à l'aide de la commande **cat** contenant le texte suivant :



Ce n'est pas grave, dans quelques jours les commandes Linux n'auront plus de secret pour moi !

- Créer un fichier appelé MonFichierConcat par concaténation des deux fichiers précédemment créés à l'aide de la commande **cat**.
- Quelle-est la commande servant à afficher le nombre de lignes d'un fichier ?
- Combien de lignes comporte votre fichier MonFichierConcat ?
- Quels sont les rôles des commandes : **head- tail- more - less**.
- Afficher les 4 premières lignes du fichier /etc/network/interfaces.

4. Manipuler les commandes de recherche : (grep/find)

4.1. La commande Grep

Rechercher les lignes contenant un terme dans un fichier

En utilisant Nano ou Vim, enregistrez ce script

```
#!/bin/bash

# Définir une variable avec le mot "Corbeau"
corbeau="Corbeau"

# Afficher le mot "Corbeau"
echo "$corbeau"

# Créer un fichier avec le mot "Corbeau" dans le nom
fichier="message_au_corbeau.txt"

# Écrire un message dans le fichier
echo "Ceci est un message pour le $corbeau !" > "$fichier"

# Afficher le contenu du fichier
cat "$fichier"

# Supprimer le fichier
rm "$fichier"
```

- Rechercher toutes les lignes contenant Corbeau dans le fichier précédent.
- Afficher le nombre de lignes que retourne la commande précédente.
- Combien d'occurrence trouvez-vous ?

Rechercher des fichiers en fonction de leur contenu

- Rechercher tous les fichiers contenant localhost dans le répertoire/etc
- Que constatez-vous en analysant le résultat ?

4.2. La commande Find

- Rechercher tous les fichiers ordinaires commençant par ho à partir du répertoire /etc

5. Créer des script avec Vim ou Nano

5.1. Entraînement :

- Via le terminal, lancez l'éditeur de texte Nano ou Vim en tapant la commande : `nano save.sh` ou `vim save.sh`

- Dans l'éditeur Nano/Vim, créer un script de sauvegarde qui copie le contenu d'un répertoire source vers un répertoire de sauvegarde.
- Rendre le script exécutable en utilisant la commande `chmod +x sauvegarde.sh` pour pouvoir l'exécuter.

5.2. Applications :

- Créer un script qui génère un journal des utilisateurs actuellement connectés à votre système Linux.
- Créer un script qui vérifie l'espace disque disponible sur votre système Linux et vous informe si l'espace est inférieur à un seuil spécifié.
- Créer un script qui parcourt le répertoire `/etc` et compte le nombre de fichiers avec l'extension `.txt`
- Créer un script qui parcourt récursivement un répertoire spécifié, affiche la taille de chaque fichier, puis calcule la taille totale du répertoire.

6. Gérer les permissions :

Application : (tout doit être fait en utilisant le Shell)

1. Préparation de l'Environnement

- Créer un groupe `devteam`.
- Créer deux utilisateurs, `dev1` et `dev2`, et les ajouter au groupe `devteam`.
- Créer un répertoire `project` dans `/home`.

2. Configuration des Permissions

- Dans `project`, créer un répertoire `config` pour stocker les fichiers de configuration.
- Changer le groupe propriétaire de `config` en `devteam`.
- Assurer que seul le propriétaire peut lire, écrire et exécuter dans `config`, tandis que les membres du groupe `devteam` peuvent uniquement lire et exécuter. Les autres utilisateurs ne devraient avoir aucun accès.

3. Validation des Permissions

- En tant que `dev1`, tenter de créer un fichier dans `config`. Cela devrait échouer.
- Changer le propriétaire de `config` à `dev1` et réessayer. Cette fois, cela devrait réussir.
- Vérifier que `dev2` peut lire mais pas écrire dans `config`.

4. Scripting de Sécurité :

- Créer un script qui automatise la vérification des permissions dans `project` pour s'assurer qu'aucun fichier ou répertoire n'est accessible en écriture par "d'autres" utilisateurs.

7. Gérer les permissions :

- Créer un répertoire `MonProjet` dans votre répertoire personnel.
- À l'intérieur de `MonProjet`, créer les sous-répertoires `Code`, `Docs`, et `Images`.
- Créer des fichiers fictifs dans chaque sous-répertoire pour simuler le contenu du projet.
- Utiliser la commande `tar` pour créer une archive de `MonProjet` sans compression.
- Compresser l'archive créée précédemment en utilisant `gzip`.
- Recréer l'archive de `MonProjet` sans compression.
- Compresser cette nouvelle archive en utilisant `bzip2`.

- Utiliser `tar` avec des options appropriées pour archiver et compresser `MonProjet` en utilisant `gzip` et `bzip2` en une seule commande pour chaque format.
 - Décompresser puis désarchiver chacun des fichiers compressés précédemment.
 - Comparer les tailles des fichiers compressés avec `gzip` et `bzip2` pour voir lequel est le plus efficace pour votre projet.
-

8. Gestion des processus :

À vous maintenant, expliquer comment gérer les processus sur les systèmes Linux, donner les commandes les plus utilisées pour contrôler les processus.

Livrables

Les lignes de commandes et les scripts des 8 parties du TP.