TP LINUX (OS UBUNTU) 4A/2024

Ce tp est une introduction à **Linux**, avec d'abord une présentation de l'utilisation du système d'exploitation **Ubuntu**, suivi de commandes bash. L'objectif principal est de tester des commandes tout en parcourant et en testant l'interface de linux. Il est nécessaire d'installer et démarrer la machine virtuelle avant de commencer le tp.

EXERCICE 1 - GUI

Il est possible de se logguer soit par l'interface graphique soit par l'interface texte (voir touches F*). Ici l'interface est graphique avant de considérer une application linux qui permet d'accéder à l'exécution de lignes de commandes sans quitter l'interface graphique.

- Logging avec l'utilisateur par défaut pour l'installation ubuntu de osboxes.
- Quels sont les logiciels installés ? Quelles fonctionnalités sont disponibles?
- **Cliquer** sur l'icône tout en haut, sous la forme d'un **dossier** ou répertoire
- Observer l'**arborescence** de fichiers, repérer vos répertoires, en particulier **Documents** et **Desktop**.
- **Cliquer** sur l' icône avec 8 points tout en bas à gauche, de nouvelles icône apparaissent.
- **Cliquer** sur l' icône pour la fenêtre de **Terminal** ou invite de commande qui exécute le code **bash**.
- **Taper** au clavier dans la fenêtre la commande **ls** puis en dessous de l'affichage la commande **ls -al**

Que constatez vous ?

EXERCICE 2 - COMMANDE

Les actions dans l'interface en ligne de commande ont une directe répercussion sur l'interface graphique puisque créer des fichier ajoute des icône tandis qu'en enlever va en supprimer.

- **Cliquer** sur l' icône avec 8 points tout en bas à gauche, de nouvelles icône apparaissent.
- **Installer** gedit en tapant dans le Terminal (fenêtre ligne de commandes) : **sudo apt install gedit** .
- **Cliquer** sur l'icône ou taper au clavier juste sous la date (après click icône avec 8 points) **gedit.**
- Ecrire une phrase dans l'éditeur de texte, cliquer sur save en donnant un nom au fichier: **texte.txt**
- Retourner à la fenêtre Terminal ouverte, taper encore au clavier la commande **ls**. Que se passe-t-il?
- Utiliser le navigateur de fichier pour retrouver le fichier crée et son contenu.

EXERCICE 3 - MAN

Les programmes utilisés en ligne de commande ont une documentation directement accessible en mode texte. Cela permet de vérifier les "options" qui sont utiles, souvent avec la syntaxe **–"option"**.

- La plupart des exercices suivants ont lieu en ligne de commande: **Aggrandir** la fenêtre **Terminal.**
- Taper au clavier la commande, **top**, à quelle programme de m.s. windows cette sortie correspond.
- Aller en annexe, et lire la documentation d'une commande en entrant: **man nom_commande.**

Le faire pour l'ensemble des premières commandes: **ls, cd, pwd, cp, mv, rm, mkdir, rmdir, cat**.
Lire les première lignes, utiliser les flêches haut/bas pour naviguer dans l'affichage. Quitter avec **Q**.

Est-ce utile ? Pourquoi ?

Comment afficher dans la fenêtre du Terminal le contenu du fichier créé juste avant, **texte.txt** à l'aide d'une des commandes disponibles?

R. Priam (2024)

EXERCICE 4 - FICHIERS ET REPERTOIRES

Les commandes permettent de nombreuses actions, telles que la création, suppresssion et modification de fichiers textes ou binaires.

- Exécuter la commande **cd Desktop**, et la commande **pwd** pour vérifier le répertoire courant
- Exécuter la commande **cd ..** et **pwd** puis revenir dans le répertoire **Desktop** précédent.
- Précédemment un fichier a été créé avec l'interface graphique, quelle commande permet de créer une nouveau fichier dans la liste en annexe?
- Utiliser la commande choisie dans la liste en annexe pour créer le fichier **essai.txt** .
- Executer les commandes vue précédemment pour afficher date création, taille, contenu.
- Le fichier **esssai.txt** est-il vide? Quelle taille lisez-vous sur le terminal d'après la commande **ls**.
- Executer la commande **echo untexte >> essai.txt** plusieurs fois, puis faire **ls -al**. Que se passe-t-il?
- Executer la commande **echo letexte > essai.txt** un seule fois, puis faire **ls -al**. Que se passe-t-il?
- Créer plusieurs fichiers avec les commandes précédentes. Puis tester les commandes, **cp, mv, rm**.
- Créer un répertoire du nom **monrépertoire** avec la commande **mkdir**. Tester **rmdir, cp, mv, rm**.

EXERCICE 5 - FICHIERS ET PERMISSION D'ACCES

A la création d'un fichier, les permissions/droits sont: -rw-rw-r-- donc il est un fichier (pas un directory/répertoire), mais le fichier est seulement avec les droits suivants **r pour read** et **w pour write**, il manque le x pour éxécutable car à la place on a **– pour le droit est non octroyé**. Donc exécuter la commande **chmod u+x nom_fichier** qui va changer **– en x** pour l'utilisateur. Les autres permissions sont pour le groupe et pour les autres (other). Les permissions permettent de protéger les données de chaque utilisateur et aussi les fichiers systèmes essentiels au bon fonctionnement du système d'exploitation linux. Par défaut le fichier est un fichier de données non exécutable donc il faut ajouter cette permission pour un script shell créé avec des commandes. Le changement de permission du fichier peut se vérifiant à l'aide de la commande **ls -al** . Une lettre **x** doit s'ajouter à droite du **w** tout à la gauche pour le user (quatrième caratère en partant de la gauche).

- Observer lors de changements de permissions d'accès (pour l'utilisateur) appliqués aux fichiers texte précédemment créés, le comportement des commandes de l'exercice précédent pour la copie, la suppression de fichiers ainsi que leur modification par ajout d'une ligne par exemple.

- Pour le cas des fichiers exécutables, cliquer à nouveau sur l'éditeur de fichier **gedit**, et créer un fichier appellé **afficherbonjour.sh** en l'enregistrant dans le répertoire **Desktop** et avec le contenu le fichier la commande **echo "bonjour"**.

- Le fichier **afficherbonjour.sh** contient une commande linux en langage bash, avec l'extension **sh** pour **shell**. Donc au lieu de taper **echo "bonjour"**, on pourrait utiliser **afficherbonjour.sh** mais le fichier a besoin d'être exécutable. Pour pouvoir executer un script enregistré dans un fichier, il faut rendre exécutable le fichier. Ajouter la permission d'exécution au fichier **afficherbonjour.sh**.

- Exécuter le script shell par la commande **./afficherbonjour.sh** où **./** s'assure que le fichier exécuté est placé dans le répertoire courant dénommé . (tester **cd .** ou **ls .** pour s'en assurer). Surtout que par défaut, le système ne cherche pas de commande exécutable dans le répertoire courant

EXERCICE 6 - VARIABLES

Le système permet de créer et accéder à des variables qui sont manipulées par le langage tandis que certains variables dites système sont réservées et contiennent des informations utiles.

- Créer la variable **a=3** et afficher le contenu de la variable par **echo $a** .
- Créer la variable **b="dutexte"** et afficher le contenu de la variable par **echo $b** .
- Lire les variables systèmes courantes, **echo $PATH**, **echo $HOME**, **echo $USER**, **echo $SHELL** **echo $PWD**, et **echo $TERM** . A quoi correspondent ces variables selon vous?

EXERCICE 7 - BOUCLE SUR REPERTOIRES ET FICHIERS

- La commande wc permet d'effectuer des comptages dans un fichier, tester **wc -m** et **wc -l** avec le fichier essai.txt précédent. Exécuter également **ls -al *.txt** puis **wc -m *.txt** et expliquer le résultat.

- Utiliser la commande **tar -xzf lesdonnées_delexo_linux.tar.gz -C fichier_destination** pour décompresser les fichiers communiqués.

Le **shell bash** présente une syntaxe très performante pour des traitement sur des fichiers contenus dans des repertoires emboités sans demander de programmer des boucles. Un exemple de commandes avec redirection de entrée/sortie est l'une des premières considérées avec le signe > au début du tp. Il est possible d'utiliser le signe | pour par exemple afficher page par page une sortie trop longue, telle que dans **find ./ | more.**

Lancer la commande et expliquer son fonctionnement en testant chaque partie séparemment (sortir par la touche **Q** si nécessaire sinon la touche **Espace** pour page suivante).

La liste de fichiers dans les sous répertoires répertoire est-elle obtenue. Modifier la commande pour afficher les nombres de mots et lignes comme précédemment mais en parcourant également les fichiers non lus avant.

- Comment lister tous les noms de fichier du répertoire et compter leur nombre de caractères et leur nombre de lignes? Tester la commande suivante.

find . -type f -name "*.txt" -exec wc -m {} \ ;

Lancer la commande et expliquer son fonctionnement en testant chaque partie séparemment.
Proposer d'autres combinaisons de commandes pour obtenir un résultat similaire.

ANNEXE I: Exemples de commandes principales pour un utilisateur linux

Commande	Description	Exemple
ls	Affiche le contenu d'un répertoire	*ls /chemin/vers/le/repertoire*
cd	Change le répertoire courant	*cd /nouveau/chemin*
pwd	Affiche le répertoire courant	*pwd*
cp	Copie des fichiers ou des répertoires	*cp source destination*
mv	Déplace ou renomme des fichiers ou des répertoires	*mv source destination*
rm	Supprime des fichiers ou des répertoires	*rm fichier*
mkdir	Crée un nouveau répertoire	*mkdir nom_du_repertoire*
rmdir	Supprime un répertoire vide	*rmdir nom_du_repertoire*
cat	Affiche le contenu d'un fichier	*cat fichier*
less	Permet une visualisation paginée du contenu d'un fichier	*less fichier*
nano	Éditeur de texte simple	*nano fichier*
vim	Éditeur de texte avancé	*vim fichier*
echo	Affiche du texte à la sortie standard	*echo "Hello World!"*
man	Affiche le manuel d'une commande	*man commande*
chmod	Ajuste les permissions d'un fichier	*chmod permissions fichier*
chown	Modifie le propriétaire d'un fichier	*chown nouveau_proprio fichier*
ps	Affiche les processus en cours d'exécution	*ps aux*
top	Affiche une vue dynamique des processus en temps réel	*top*
kill	Termine un processus en utilisant son PID	*kill PID*
killall	Termine tous les processus associés à un nom	*killall nom_processus*
df	Affiche l'utilisation de l'espace disque	*df -h*
du	Affiche l'utilisation de l'espace disque par dossier	*du -h dossier*
free	Affiche la mémoire disponible	*free -m*
wget	Télécharge des fichiers depuis le Web	*wget URL*
tar	Archive et extrait des fichiers	*tar -cvf archive.tar fic1 fic2*
gzip	Compresse des fichiers	*gzip fichier*
uname	Affiche des informations sur le système	*uname -a*
ifconfig	Affiche la configuration réseau	*ifconfig*
ping	Envoie des paquets à une adresse IP pour tester la connect.	*ping adresse_ip*
grep	Recherche un motif dans un fichier ou sortie de commande	*grep motif fichier*
find	Recherche des fichiers dans un répertoire	*find /chemin -name fichier*
sed	Éditeur de flux (pour la recherche/transformation de texte)	*sed 's/motif1/motif2/g' fichier*
awk	Traitement de texte et extraction de données	*awk '{print $1}' fichier*
ln	Crée des liens entre les fichiers	*ln -s source lien*
chmod	Ajuste les permissions d'un fichier	*chmod permissions fichier*
chown	Modifie le propriétaire d'un fichier	*chown nouveau_proprio fichier*
sudo	Exécute des commandes avec des privilèges root	*sudo commande*
touch	Création d’un fichier	*touch fichier*
tree	Afficher l’arborescence des fichiers	*tree ./*
wc	Comptage de caractère et lignes dans un fichier	*wc -m fichier*
which	Affiche le chemin d’une commande dans les fichiers linux	*which commande*

ANNEXE II: Exemples de commande administrateur - super utilisateur - root.
Certain appels peuvent affecter le fonctionnement de linux et les données utilisateur.

Commande	Description	Exemple
apt/yum	Gère les paquets logiciels (selon la distribution)	*apt install nom_paquet*
chmod	Ajuste les permissions d'un fichier	*chmod permissions fichier*
chown	Modifie le propriétaire d'un fichier	*chown nouveau_user*
df	Vérifie l'utilisation de l'espace disque	*df -h*
du	Montre l'utilisation de l'espace disque par dossier	*du -h dossier*
fdisk	Gère les partitions du disque en mode texte	*fdisk /dev/sdX*
journalctl	Affiche les journaux du système	*journalctl*
passwd	Change le mot de passe d'un utilisateur	*passwd nom_utilisateur*
reboot	Redémarre le système	*reboot*
shutdown	Éteint le système	*shutdown -h now*
sudo	Exécute des commandes avec des privilèges root	*sudo commande*
systemctl	Contrôle le système init et ses services sous systemd	*systemctl start nom_service*
useradd	Ajoute un nouvel utilisateur	*useradd nom_utilisateur*
userdel	Supprime un utilisateur	*userdel nom_utilisateur*

ANNEXE III: Exemples autres commandes ou autres structures syntaxiques

Commande	Description
for var in liste; do something done	Boucle pour var parcourant les valeurs de la liste, tableau ou autre structure pouvant être itérée. La partie de code something est exécuté pour chaque valeur différente de var.
while test; do something done	Boucle exécutant la partie de code something jusque la condition d'arrêt renseignée dans la partie test.

**if test1; then
  something 1
elif test2; then
  something 2
else
  something 3
fi**
	Execute something1 si la condition test1 est vrai, sinon exécute something2 si la condition test2 est vrai ou sinon something3. Les parties elif et else sont en option, et davantage de elif + somethingk peuvent être ajoutés pour prendre en compte d'autres tests à vérifier.

**case $var in
  test1) something1;;
  test2) something2;;
  ...
  testn) somethingn;;
esac**
	Trouve le test vrai testk pour $var et exécute la partie de code somethingk qui lui correspond.

(expression)	Permet de regrouper des commandes ou rediriger les entrées et sorties de plusieurs commandes.
cmd1 ; cmd2	Permet de séparer deux commandes sur une même ligne.