

LINUX - Initiation à l'utilisation des systèmes

SOMMAIRE

TP-1-Découverte du shell.....	3
TP-2-Manipulation des fichiers et des répertoires....	4
TP-3-Manipuler le contenu des fichiers.....	5
TP-4-Manipuler les commandes de recherche.....	6
TP-5-Utilisation de vim.....	8
TP-6-Script et gestion des permissions.....	9
TP-7-Archivage et compression.....	10
TP-8-Redirections.....	11
TP-9-Gestion des processus.....	13

TP-1-Découverte du shell

Changer de mot de passe, fermer puis rouvrir la session pour vérifier la prise en compte du mot de passe

"passwd nom_utilisateur"

Quel utilisateur est connecté ?

"UTIL: kaly"

Depuis quelle heure ?

"LOGIN : 11:53"

Quel est le temps d'activité de cet utilisateur ?*"IDLE : 7:48"*

```
kaly@Debian666:~$ w
11:56:29 up 7 min, 1 user, load average: 0,23, 0,32, 0,18
UTIL.    TTY      DE              LOGIN@  IDLE   JCPU   PCPU   QUOI
kaly     tty2      tty2            11:53   7:48   0.02s  0.02s  /usr/libexec/gnome-session-binary
kaly@Debian666:~$
```

Quel est le numéro de processus de votre session ?

"PID : 2516"

"ps" pour tous les processus

"echo \$\$" \$\$ est l'ID de processus (PID) du script lui-même.

```
kaly@Debian666:~$ ps
  PID TTY          TIME CMD
 2516 pts/0        00:00:00 bash
 2560 pts/0        00:00:00 ps
kaly@Debian666:~$ echo $$
2516
kaly@Debian666:~$
```

A quel groupe(s) appartenez-vous ? Quels sont vos identifiants d'utilisateur (UID) et de groupes (GID) ?

"id kaly"

| Groupe(1000) | UID(1000) | GID(1000) |

```
kaly@Debian666:~$ id kaly
uid=1000(kaly) gid=1000(kaly) groupes=1000(kaly),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),100(users),106(netdev),111(bluetooth),113(lpadmin),116(scanner)
```

Afficher la date et l'heure

"date"

TP-2-Manipulation des fichiers et des répertoires

Qu'est-ce que le répertoire courant ?

Le répertoire courant est le répertoire dans lequel vous vous trouvez actuellement lorsque vous utilisez le terminal.

Afficher le répertoire courant

"ls"

Quelle-est la spécificité de ce répertoire ?

"La spécificité de ce répertoire dépend de l'endroit où vous vous trouvez dans la hiérarchie du système de fichiers"

Existe-t-il des fichiers et/ou des répertoires dans votre répertoire courant ? Le(s)quel(s) ?

"Bureau Documents Form Images Modèles Musique Public Téléchargements Vidéos"

Création de répertoire et de fichier

"mkdir MonRepertoire"

"cd MonRepertoire"

"touch MonFichier"

Afficher les informations détaillées des répertoires et des fichiers

"ls -l MonRepertoire"

"ls -l MonFichier"

Rechercher des fichiers particuliers

"find /etc -type f -name '.conf'"*

TP-3-Manipuler le contenu des fichiers

Créer un fichier appelé MonDeuxiemeFichier dans votre home à l'aide de la commande touch

"touch MonDeuxiemeFichier"

Quels-sont les éditeurs de texte en CLI principaux de Linux ?

"Nano / Vim"

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

"touch MonTroisiemeFichier.txt"

"echo "Ce n'est pas grave, dans quelque jours les commandes Linux n'auront plus de secret pour moi !" >> MonTroisiemeFichier.txt"

Créer un fichier appelé MonFichierConcat par concaténation des deux fichiers précédemment créés à l'aide de la commande cat

"cat MonDeuxiemeFichier.txt MonTroisiemeFichier.txt > MonFichierConcat.txt"

Quelle-est la commande servant à afficher le nombre de lignes d'un fichier ?

"wc -l MonFichierConcat.txt"

Combien de lignes comporte votre fichier MonFichierConcat ?

"3 Lignes"

Quel est le rôle de la commande head ?

La commande head est utilisée pour afficher les premières lignes d'un fichier. Par défaut, elle affiche les 10 premières lignes d'un fichier, mais vous pouvez spécifier un nombre différent en utilisant l'option -n.

"head -n 5 MonFichierConcat.txt"

Quel est le rôle de la commande tail ?

La commande tail est l'opposée de head. Elle affiche les dernières lignes d'un fichier.

Par défaut, elle affiche les 10 dernières lignes, mais vous pouvez également spécifier un nombre différent avec l'option -n.

"tail -n 7 MonFichierConcat.txt"

Quel est le rôle de la commande more ?

La commande more est utilisée pour afficher le contenu d'un fichier page par page.

Elle permet de parcourir un fichier texte en utilisant les touches de défilement (comme la touche Espace pour passer à la page suivante).

Cependant, more ne permet pas de revenir en arrière dans le fichier.

Pour afficher le contenu de mon_fichier.txt avec more, exécutez simplement "more mon_fichier.txt"

Quel est le rôle de la commande less ?

Elle permet de parcourir le fichier vers l'avant et vers l'arrière.

Vous pouvez utiliser les touches fléchées, la barre d'espace, la touche B pour revenir en arrière, et la touche Q pour quitter. Pour afficher le contenu de mon_fichier.txt avec less, utilisez :

"less mon_fichier.txt"

Quelle est la différence avec more ?

More est plus basique et ne permet que le défilement vers l'avant, tandis que less offre plus de fonctionnalités pour naviguer dans le fichier.

Afficher les 4 premières lignes du fichier /etc/network/interfaces

"head -n 4 /etc/network/interfaces"

TP-4-Manipuler les commandes de recherche

Copier le fichier TP_recherche dans votre répertoire personnel et nommez-le TP_recherche_login (en remplaçant « login » par votre propre login).

`"cp 1231956083611013264.txt TP_recherche_kaly"`

Rechercher les lignes contenant un terme dans un fichier sans tenir compte de la casse

`"grep -i "corbeau" TP_recherche_kaly"`

Afficher le nombre de lignes que retourne la commande précédente

*"Maître Corbeau, sur un arbre perché,
Et bonjour, Monsieur du Corbeau.
À ces mots, le Corbeau ne se sent pas de joie ;
Le Corbeau honteux et confus"*

Combien d'occurrence trouvez-vous ?

4

Rechercher des fichiers en fonction de leur contenu

`"grep -r "localhost" /etc"`

"L'option -r dans la commande grep signifie "récuratif". Lorsque vous utilisez -r, grep parcourt récursivement tous les sous-répertoires à la recherche du terme spécifié dans les fichiers. Cela signifie qu'il explorera également les sous-dossiers du répertoire spécifié, pas seulement les fichiers directement dans ce répertoire"

Que constatez-vous en analysant le résultat ?

"Permission non accordée" sur la plupart des fichiers dans le répertoire

A présent, rechercher tous les fichiers contenant localhost dans le répertoire/etc sans inclure les répertoires

`"grep -l "localhost" /etc/*"`

Rechercher des fichiers en fonction de leur nom

`"find /etc -type f -name 'ho*'"`

Rechercher tous les répertoires depuis votre répertoire personnel

"find /home/kaly"

"find /home/kaly -type d" (-d pour les dossiers)

BONUS : tentez de réaliser cette action en désignant votre répertoire personnel de 4 manières différentes

TP-5-Utilisation de vim

Installation de vim

"sudo apt install vim"

Récupérer le fichier du TP puis l'ouvrir

"cp tp-5-utilisation-de-vim-6536245fd1c0b952883659.pdf TP_vim_kaly.pdf"

Ouvrir le fichier à l'aide de vim

"vim TP_vim_kaly.pdf"

Rechercher « Patrick Sanchez » dans le fichier

"grep -i "Patrick Sanchez" TP_vim_kaly.txt"

Quels sont les raccourcis pour naviguer entre les occurrences ?

utilisez les touches fléchées haut et bas

Combien il y a-t-il d'occurrence de cette chaîne de caractères ?

"7"

Modifier la première occurrence : se déplacer sur le k de Patrick et le remplacer par un e pour obtenir « Patrice Sanchez »

"r" "e"

Sans utiliser les flèches directionnelles, se placer en début de fichier

"gg"

Sans utiliser les flèches directionnelles, se placer en fin de fichier

"G"

Chercher Frédéric Poirier puis insérer votre prénom et votre nom

"o"

Suppression de ligne et de caractère

"dd"

Annuler cette action à l'aide du bon raccourci

"u"

Supprimer le deuxième e de « Philippee » à l'aide du bon raccourci

"x"

Remplacement de mot

"cw"

Copier/coller

"yy" - "p"

Rechercher et remplacement

":%s/motif/remplacement/g"

Enregistrer et quitter

":wq"

TP-6-Script et gestion des permissions

Que veut dire #! /bin/bash ?

`#! /bin/bash` est ce qu'on appelle un "shebang" ou "hashbang". C'est une ligne ajoutée au début de certains fichiers exécutables sur les systèmes Unix/Linux, qui indique au système d'exploitation quel interpréteur doit être utilisé pour exécuter le script.

Plus précisément, le shebang `#! /bin/bash` signifie:

- `#!` est une séquence spéciale qui indique au système qu'il s'agit d'un shebang.
- `/bin/bash` est le chemin absolu vers l'interpréteur Bash (Bourne-Again SHell).

C'est l'interpréteur de commandes le plus courant sur les systèmes Unix/Linux.

Comment exécuter ce script ?

“/bin/bash Bonjour”

Quelles permissions ont été définies par défaut ?

L'umask (user file-creation mode mask)

Généralement, l'umask par défaut est 0022.

Pour les fichiers, les permissions par défaut sont:

- Lecture/écriture pour le propriétaire (644 en notation octale)*
- Lecture pour le groupe et les autres*

Quel propriétaire et groupe propriétaire a été défini par défaut ?

Le propriétaire (owner) par défaut du fichier est votre utilisateur courant, celui avec lequel vous êtes connecté et avez créé le fichier.

Le groupe propriétaire (group owner) par défaut du fichier est votre groupe primaire, généralement le même que le nom d'utilisateur sauf si vous appartenez à d'autres groupes secondaires.

Quelles sont les commandes pour changer

L'utilisateur propriétaire

“chown nouvel_utilisateur fichier”

Le groupe propriétaire

“chgrp nouveau_groupe fichier”

Les permissions

“chmod permissions fichier”

Attribuer les permissions adéquates pour permettre au propriétaire d'exécuter ce script

“chmod u+x Bonjour”

Supprimer les permissions aux membres du groupe ainsi qu'à tous les autres sur le fichier

“chmod go-rwx Bonjour”

TP-7-Archivage et compression

Réaliser une sauvegarde de votre répertoire personnel et placer là dans /tmp en utilisant votre login pour le nommage du fichier

`"tar czvf /tmp/kaly.tar.gz."`

`"tar -cvzf /tmp/sauvegarde_$KALY.tar.gz /^Cme/kaly/Form/"`

Afficher la taille (lisible) de l'archive précédemment créée et noter cette taille

`"ls -lh /tmp/kaly.tar.gz"`

`"-rw-r--r-- 1 kaly kaly 17M 23 avril 10:58 /tmp/kaly.tar.gz"`

Créer un fichier vide dans votre répertoire personnel

`"touch fichier_vide.txt"`

Mettre à jour l'archive précédemment créée

`"tar -rvf archive.tar.gz fichier_vide"`

Réaliser une nouvelle sauvegarde de votre répertoire personnel en la compressant avec bzip2 (même consignes pour le nommage mais en utilisant l'extension .tar.bz2)

`"tar cjvf /tmp/mon_login.tar.bz2."`

Afficher la taille (lisible) de l'archive précédemment créée et noter cette taille

`"ls -lh /tmp/kaly.tar.gz"`

`"-rw-r--r-- 1 kaly kaly 17M 23 avril 10:58 /tmp/kaly.tar.gz"`

Dans votre répertoire personnel, créer un répertoire nommé Restauration

`"mkdir restauration"`

Restaurer la dernière sauvegarde dans ce répertoire

`"tar -xvzf /home/kaly/Form_$USER.tar.gz -C ~/restauration"`

Créer une nouvelle sauvegarde de votre répertoire personnel en excluant le répertoire Restauration

`"tar -czvf /home/kaly/Form_$USER.tar.gz-exclude=/restauration"`

TP-8-Redirections

Afficher tous les processus en cours pour tous les utilisateurs

“ps aux”

Répéter l'action précédente et n'afficher que les lignes comportant le terme "ssh" en utilisant un pipe

“ps aux | grep ssh”

A partir de la commande précédente, afficher uniquement le nombre de lignes

“ps aux | grep ssh | wc -l”

Saisir les prénoms des apprenants de votre groupe en tapant entrée à chaque fois, Que se passe-t-il ?

“Ils sont triée par ordre alphabétique”

Créer un fichier nommé groupe et insérer dans ce fichiers 10 prénoms • Utiliser la commande sort en spécifiant en entrée le fichier groupe précédemment créé

“sort exo7.txt”

Rechercher les fichiers ordinaires dont le nom se termine par .conf à partir de la racine (cette commande servira de base pour la suite du TP)

“find / -type f -name ".conf”*

Adapter la commande précédente afin de ne pas afficher les erreurs à l'écran à l'aide de la redirection du flux de sortie d'erreur

“find / -type f -name ".conf" 2>/dev/null”*

Adapter la commande précédente afin de stocker le résultat dans un fichier nommé `resultats1_fichiers_conf.txt` dans votre répertoire personnel

`"find / -type f -name "*.conf" 2>/dev/null ~/resultats1_fichiers_conf.txt"`

Vérifier le contenu du fichier `resultats1_fichiers_conf.txt`

`"cat ~/resultats1_fichiers_conf.txt"`

Adapter la commande précédente afin d'afficher le résultat à l'écran tout en le stockant dans un fichier nommé `resultats2_fichiers_conf.txt` dans votre répertoire personnel

`"find / -type f -name "*.conf" 2>/dev/null | tee ~/resultats2_fichiers_conf.txt"`

Vérifier le contenu du fichier `resultats2_fichiers_conf.txt`

`cat ~/resultats2_fichiers_conf.txt`

TP-9-Gestion des processus

Quel est le rôle des commandes suivantes et quelles sont leurs principales options ?

ps : La commande ``ps`` (pour "process status") est utilisée pour afficher des informations sur les processus en cours d'exécution sur le système.

Principales options

- ``-e`` : Affiche tous les processus.
- ``-f`` : Affiche une sortie formatée incluant plus de détails sur les processus.
- ``-u <utilisateur>`` : Affiche les processus de l'utilisateur spécifié.
- ``-aux`` : Affiche une liste détaillée de tous les processus en cours, y compris ceux des autres utilisateurs.

pstree : La commande ``pstree`` affiche les processus sous forme d'arbre hiérarchique, montrant les relations de parenté entre les processus.

Principales options

- ``-p`` : Affiche les identifiants de processus (PIDs) dans la sortie.

``-u`` : Affiche les propriétaires des processus.
``-G`` : Affiche les groupes propriétaires des processus.

top : Est une commande interactive qui affiche dynamiquement les informations sur les processus et l'utilisation des ressources système en temps réel.

Principales options

``-d <délai>`` : Définit le délai de rafraîchissement en secondes.
``-n <nombre>`` : Définit le nombre de mises à jour à afficher avant de quitter.
``-u <utilisateur>`` : Affiche les processus de l'utilisateur spécifié.
``-p <PID>`` : Affiche les informations uniquement pour le PID spécifié.

pgrep : La commande ``pgrep`` recherche les processus en cours d'exécution basés sur leurs noms et renvoie les PIDs correspondants.

Principales options

``-l`` : Affiche le nom du processus en plus de son PID.
``-u <utilisateur>`` : Affiche les processus de l'utilisateur spécifié.
``-x`` : Recherche uniquement les processus dont le nom correspond exactement à celui spécifié.

Quel est le rôle des commandes suivantes et quelles sont leurs principales options ?

kill : La commande ``kill`` est utilisée pour envoyer un signal à un processus, généralement dans le but de le terminer. Par défaut, ``kill`` envoie le signal TERM (15), mais d'autres signaux peuvent être spécifiés.

Principales options

``-<signal>`` : Spécifie le signal à envoyer au processus.
Par exemple, ``-9`` envoie le signal KILL, qui force le processus à se terminer immédiatement.

time : La commande ``time`` est utilisée pour mesurer le temps d'exécution d'une commande ou d'un processus. Elle affiche le temps écoulé, le temps CPU utilisateur et le temps CPU système.

Principales options

n'a pas beaucoup d'options spécifiques, mais elle peut être utilisée avec n'importe quelle commande pour mesurer son temps d'exécution.

Par exemple

``time <commande>`` : Exécute la commande spécifiée et affiche les informations de temps.
``-p`` : Utilise un format de sortie plus précis, compatible avec le shell POSIX.

Comment lancer une commande en arrière-plan ?

“commande &”

Qu'est-ce qu'un job ? Quel-est le rôle de la commande jobs ?

“Fait référence à une tâche ou à un processus en cours d'exécution dans un shell

Le rôle de la commande “jobs” est de fournir des informations sur les jobs en cours d'exécution dans le shell.

Affichage des jobs : La commande “jobs” affiche une liste des jobs en cours d'exécution dans le shell actuel, ainsi que leur numéro de job, leur statut et leur commande associée

Gestion des jobs : En plus de simplement afficher les jobs, la commande “jobs” peut également être utilisée pour gérer les jobs. Par exemple, vous pouvez utiliser “fg” (foreground) pour amener un job en arrière-plan au premier plan, ou “bg” (background) pour démarrer un job en arrière-plan

Lister les processus en cours finissant par un « d »

“ps -ef | grep 'd\$’”

Lister les processus en cours finissant par un « b »

“ps -ef | grep 'b\$’”

Identifier le PID de votre session SSH

“pgrep ssh”

Terminer ce processus

“kill 1540”

Modifier la commande suivante afin de n'afficher que son temps de traitement

“time find / -name '' > /dev/null”*

Exécuter ce script en arrière-plan

“bash bonjour &”

Afficher le job avec son PID

“jobs”

Arrêter le job avant sa fin

“kill %1”