debian

Le système

Le système - Connexion

- Travailler sous Linux, même en dehors du réseau, implique une connexion au système. Une session monoposte n'est jamais anonyme et le processus d'authentification est classique:
  - nom de l'utilisateur
  - Mot de passe

- Authentifiés, vous disposez des ressources du système selon vos permissions (les droits de fichiers) que l'administrateur (root) vous a accordé.

Le système – multi-Connexions

- Il est possible de se connecter plusieurs fois sur une même machine sous des identités différentes puisque Linux est un environnement multi-utilisateurs.
- Pour cela, on peut ouvrir des consoles textes par l'intermédiaire des touches **ALT+F1 à ALT+F6**
- Cette combinaison de touches est très utilise lors des TP, elle permet de changer d'utilisateur sans avoir à se déconnecter.

Le système – déconnexion et arrêt utilisateur

- Pour se déconnecter de la console, entrez « **exit** », « **logout** » ou **CTRL+D**. cela relance l'attente de login.
- En cas de coupure brutale, le système effectuera des réparation au prochain démarrage, à l'aide de l'utilitaire « fsck » avant de procéder à l'initialisation du système.
- Si un utilisateur quelconque peut ouvrir une session, il ne peut arrêter la machine (ou serveur), cela est réservé à l'utilisateur root.

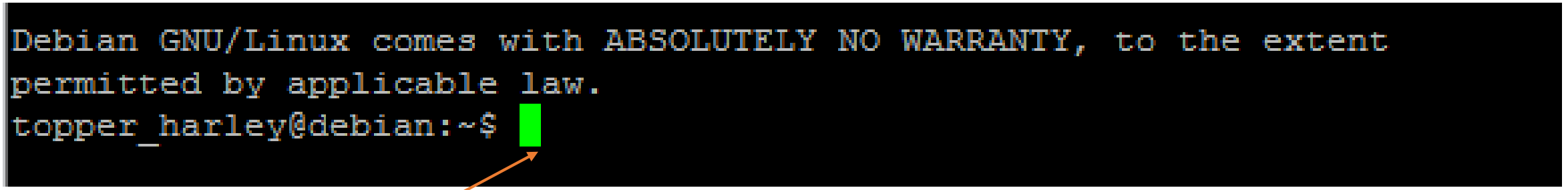
Arrêt immédiat	halt	shutdown –h now
Arrêt différé	shutdown –h <nb_min>	
redémarrage	reboot	shutdown –r [<nb_min>|<now>] ou ctrl+alt+suppr

debian

*Premier contact avec le **SHELL***

Notions de shell

- Le shell est un programme qui va faire le lien entre le noyau UNIX et l'utilisateur
- Le shell est un interpréteur de commandes qui invite l'utilisateur à saisir une commande et la fait ensuite exécuter



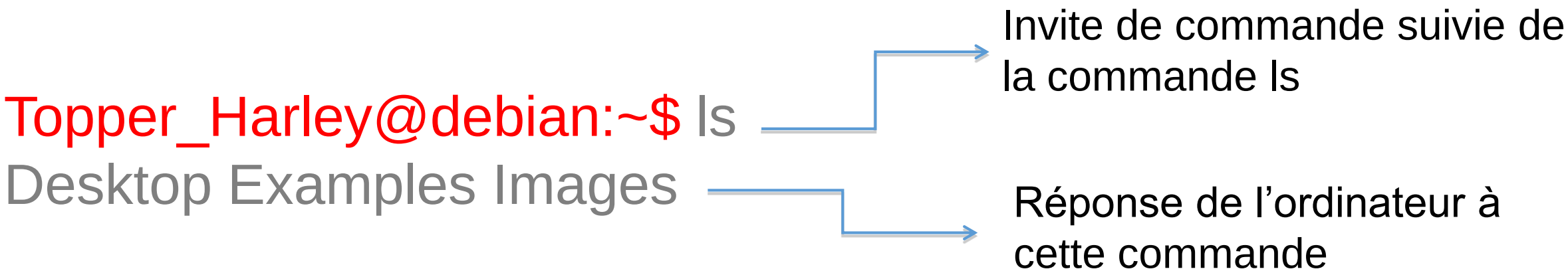
le shell est prêt à recevoir des commandes entrées au clavier

l'invite de commande

Soit le prompt **topper_harley@debian:~$**

- **Topper_harley**: le premier élément est votre nom.
- **debian**: nom de l'ordinateur sur lequel vous travaillez.
  La ligne d'invite de commandes se lit donc « Topper Harley » chez la machine « debian ». En d'autres termes, je suis identifié en tant que Topper Harley sur la machine debian.
- **:** est un séparateur
- **~** est le dossier dans lequel vous vous trouvez actuellement. Le symbole ~ signifie que vous êtes dans votre dossier personnel, ce qu'on appelle le "Home" sous Linux. C'est l'équivalent du dossier "Mes documents" de Windows.
- **$** : ce dernier symbole est très important, il indique votre niveau d'autorisation sur la machine. Il peut prendre 2 formes différentes :
  - $: signifie que vous êtes en train d'utiliser un compte utilisateur "normal",
  - # : signifie que vous êtes en mode super-utilisateur (root)

L’invite de commandes



Cela signifie que le répertoire actuel est constitué de 3 dossiers : Desktop Examples et Images. En général, le système colore les éléments pour que l'on puisse distinguer facilement les dossiers des fichiers. Si vous n'avez aucune réponse, c'est que vous êtes dans un dossier qui ne contient aucun fichier ou dossier.

Les paramètres

- Les paramètres sont des options que l'on écrit à la suite de la commande.
  Topper_Harley@debian :~$ Commande <Paramètres>
- **Exemples de paramètres:**
  Topper_Harley@debian :~$ **ls -a**
- Pour connaitre les paramètres d'une commande, vous devez utiliser l'instruction suivante:

  Topper_Harley@debian :~$ man <*Commande*>
- Soit par exemple: Topper_Harley@debian :~$ man *useradd*

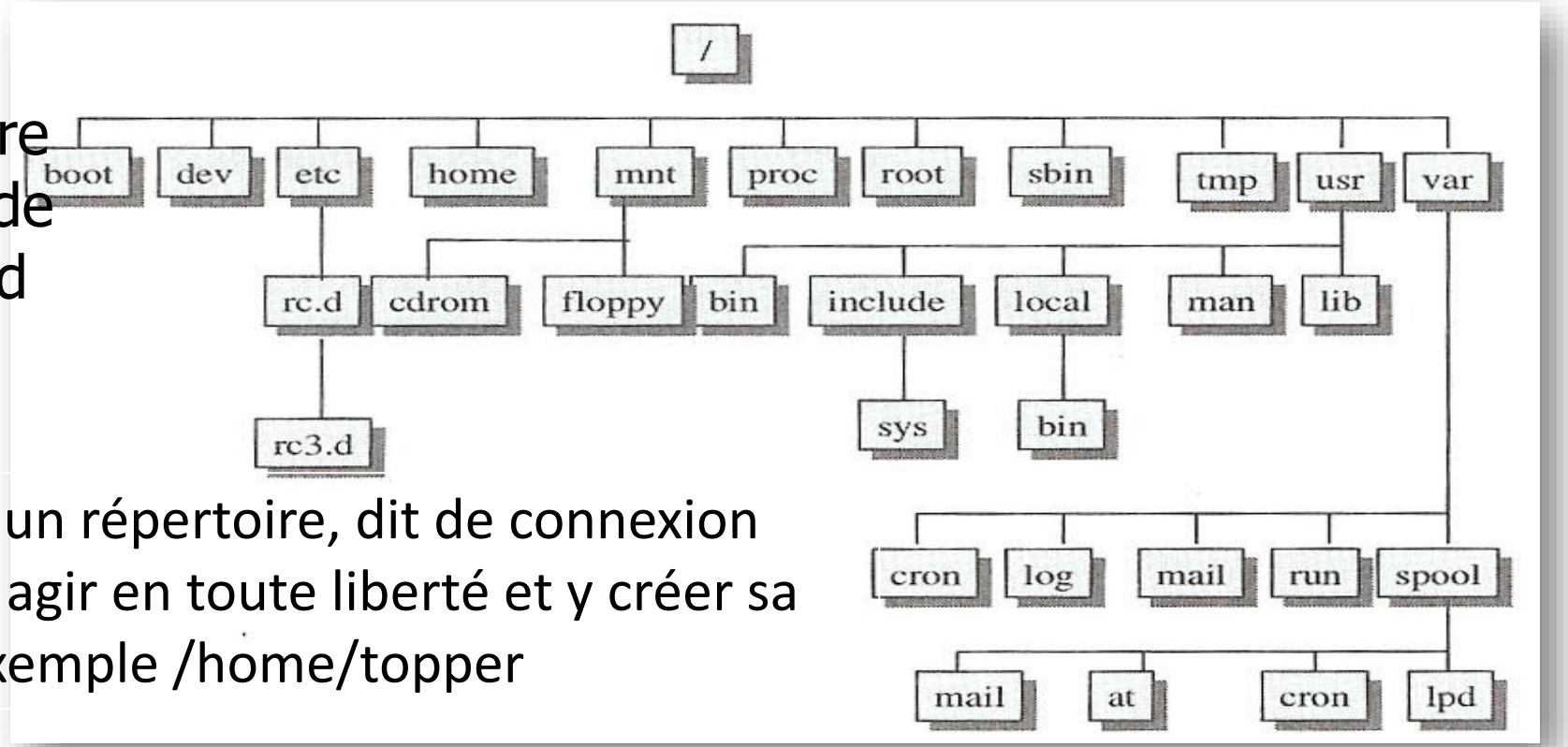
  Appuyez sur « q » pour quitter votre page man

Historique des commandes

- Pour retrouver les commandes que l'on a déjà tapé, sans devoir les réécrire à nouveau, on utilise les flèches haut ou bas de notre clavier.
- Si on désire visualiser les commandes qui remontent à très longtemps, inutile de se forcer à taper sur la flèche haut, pour cela, utilisez la commande history.
- Faire un #man history pour connaitre les paramètres de cette commande!

Organisation des dossiers

- Une racine est le point le plus haut dans l'arborescence d'un disque dur.
- Sous Windows, il y a une racine par unité de disque (ex: c:\) . Sous Linux, les unités sont montées dans un dossier, il n'y a donc qu'une seule racine (/).
- Un cd-rom pourra donc être accessible par la commande **#cd /mnt/cdrom** si le cd à été monté dans /mnt/cdrom



- Chaque Utilisateur possède un répertoire, dit de connexion (home directory), ou il peut agir en toute liberté et y créer sa propre arborescence, par exemple /home/topper

Arborescence des fichiers

/ : Répertoire Racine-> contient les répertoires principaux.

/bin : commandes de base du système-> contient des exécutables essentiels au système, employés par tous les utilisateurs (exemple: ls, rm, cp, chmod, mount,…).

/sbin: contient les binaires système essentiels (par exemple la commande useradd/d'administration du système./ commandes dangereuses réservées à l'administrateur.

/etc : contient les commandes et les fichiers nécessaires à l'administrateur du système ->fichiers passwd, group, inittab, interfaces, fichiers de configuration de l'ordinateur et des services réseaux.

/boot: contient les fichiers statiques (le noyau vmlinuz, sauvegardes secteurs de demarrage, …) du chargeur de demarrage.

/dev : contient les points d'entrée des périphériques disponibles

/lib: contient des bibliothèques partagées essentielles au système lors du démarrage

/mnt : contient les points de montage des partitions temporaires(cd-rom, disquette…)

/opt: contient des pakages d'applications supplémentaires (ex: staroffice, java, …)

ARBORESCENCES DES FICHIERS

/root: homedirectory de root

/tmp: contient les fichiers temporaires

/usr: (Hiérarchie secondaire)/ programmes utilisateurs ordinaires, non essentiels au systèmes: /usr/lib, /usr/bin…

/home: contient les dossiers de travail des utilisateurs. Ex:/home/Topper Harley). Il s'agit du profil utilisateur (correspond à documents & settings sous Windows)

/var: contient des données variables liées à la machine (fichiers d'impression, traces de connexions http, log (/var/log).

COMMANDES DE BASES : Répertoires

La commande ls

- La commande ls (list): Affiche la liste des fichiers dans le répertoire courant, en ordre alphanumérique, sauf ceux qui commencent par le caractère “.” (sauf les fichiers cachés)
- « pour cacher un fichier sous linux, on le précède d’un « . »)
- ls [options] [noms]

noms du ficher qu’on veut lister. Si ce paramètre est absent, ls affichera les fichiers du répertoire courant.

- Exemple : ls -l /home/Topper Harley

COMMANDES DE BASES : Répertoires

- **La commande ls**
  - **ls a** («all»: tous) Affiche tous les fichiers (y compris les fichiers .*)
  - **ls l** (long) Affichage en format long (type, date, taille, propriétaire, permissions)
  - **ls t** (temps) Affiche les fichiers les plus récents en premier.
  - **ls d** (directory) Affiche du nom du répertoire et non de son contenu.
  - **ls S** (“size”: taille) Affiche les fichiers les gros en premier.
  - **ls r** («reverse»: inversé) Affiche en ordre inverse.
  - **ls ltr** (les options peuvent être combinées) Format long, les fichiers les plus récents à la fin.
  - **ls h:** afficher la taille en Ko, Mo, Go...

COMMANDES DE BASES : Répertoires

Commande cd

- La commande cd (change directory) permet de changer le répertoire de travail (permet de naviguer dans les répertoires)
- Le système de déplacement dans les dossiers est strictement identique au système Windows (déplacements relatifs et absolus).
- De ce fait, les 2 types de déplacement ne sera pas abordé dans ce cours.
- Exemple: cd [répertoire] *Si répertoire n'est pas précisé, alors le nouveau répertoire sera le répertoire de connexion*
- Si on veut se positioner dans le répertoire /etc, on fera ceci:

```
Topper Harley@debian :~$ cd /etc
Topper Harley@debian :/etc$
```

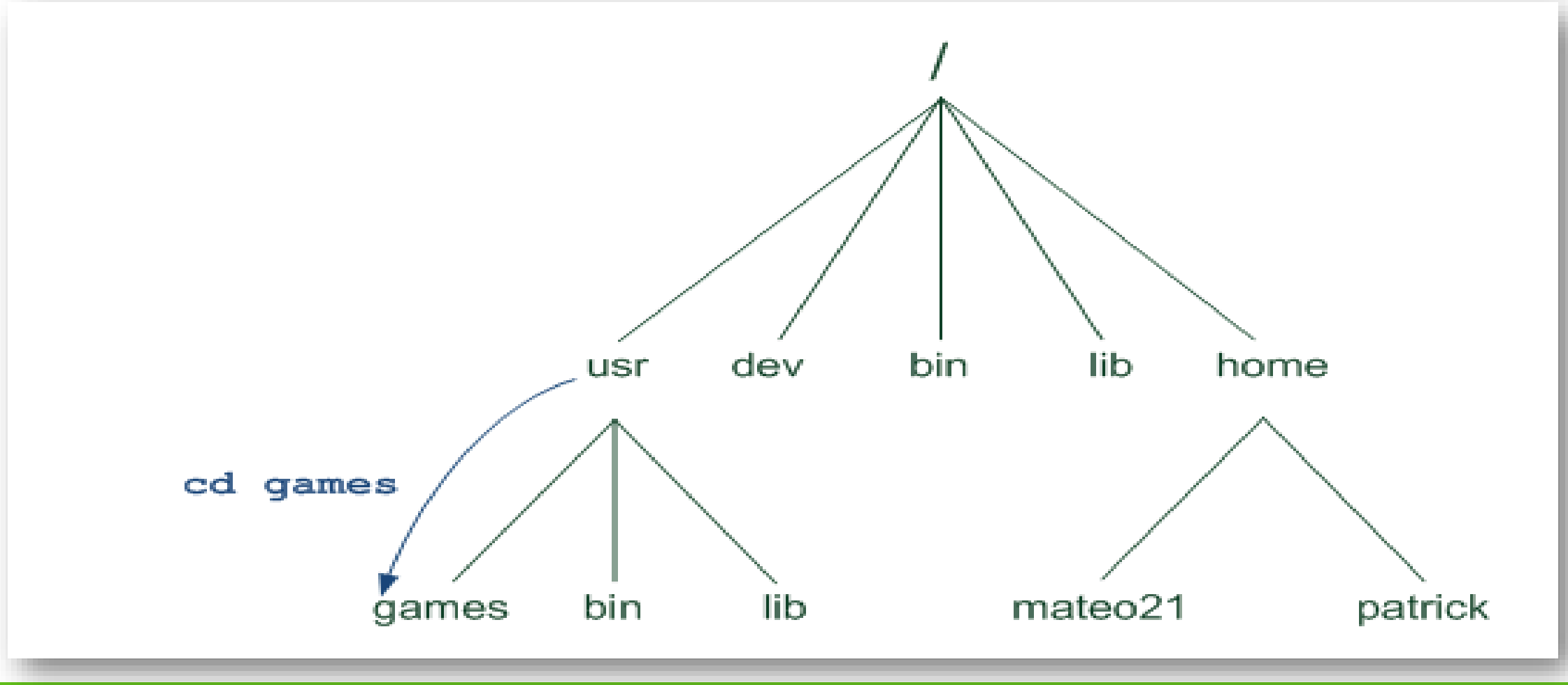
COMMANDES DE BASES : Répértoires

Commande cd

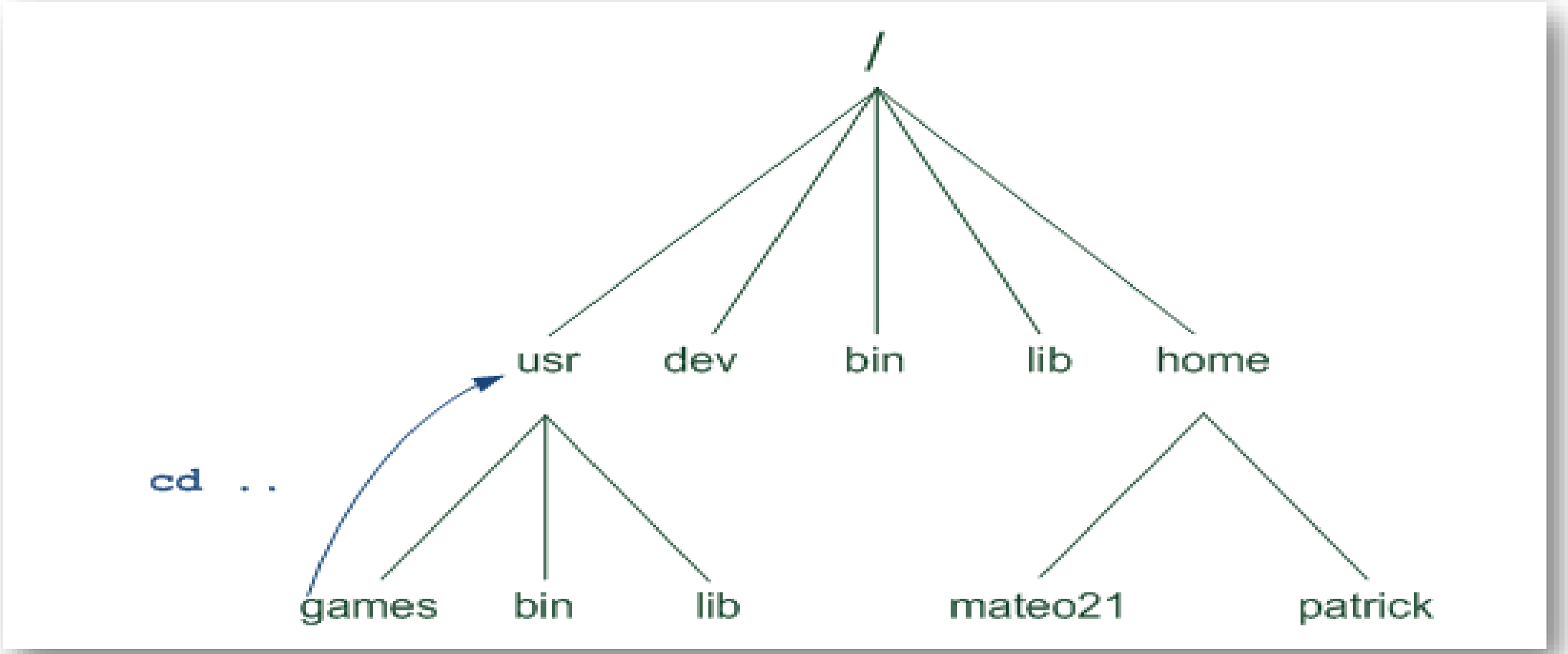
- **cd** ou **cd ~** ou **cd $home**: Permet de se placer dans le répertoire personnel de l'utilisateur (home)
- **cd ..** : Ramène dans le répertoire parent (remonte d'un répertoire dans l'arborescence)
- **cd <rep>** : va dans le répertoire rep à partir de l'endroit où vous êtes (déplacement relatif). Peut-être combiné à **cd ..**
- **cd /<rep>:** va dans un répertoire en passant par la racine (déplacement absolu)

Opérations de base sur les répertoires

- **Les chemins relatifs** :Un chemin relatif est un chemin qui dépend du dossier dans lequel vous vous trouvez.



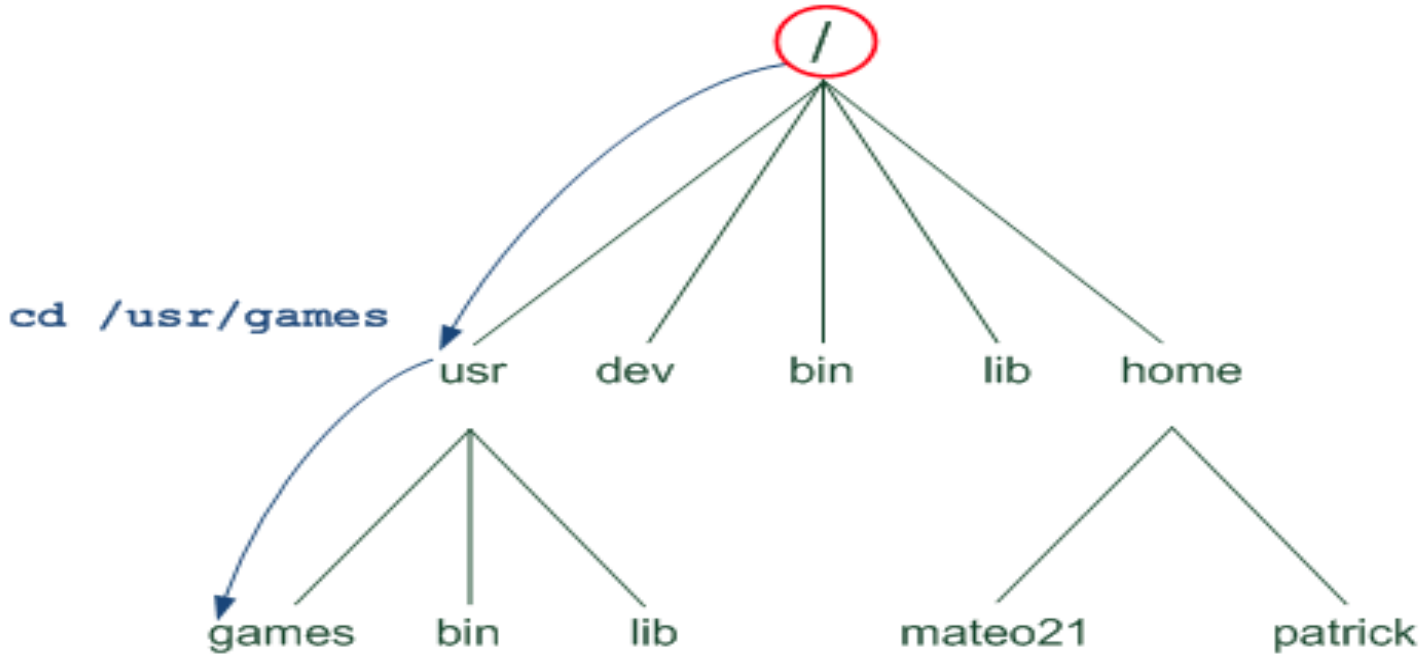
Opérations de base sur les répertoires



Opérations de base sur les répertoires

Les chemins absolus :les chemins absolus fonctionnent quel que soit le dossier dans lequel on se trouve. Un chemin absolu est facile à reconnaître : il commence toujours par la racine (/). Vous devez faire ensuite la liste des dossiers dans lesquels vous voulez entrer.

```
mateo21@mateo21-desktop:~$ cd /usr/games
mateo21@mateo21-desktop:/usr/games$
```



Opérations de base sur les répertoires

- **Commande pwd**

- La commande **pwd** (**print working directory**): Affiche le répertoire courant ("répertoire de travail »)
- La commande **pwd**): peut etre utilisée lorsqu'on désire obtenir un chemin absolu sur le répertoire courant.

- Cette commande est également utilisée par les scripts pour déterminer le répertoire à partir duquel ils sont exécutés.
- *La commande pwd n'accepte pas d'option.*

Exemple :

Topper Harley@debian :~$ pwd
/home/Topper Harley

COMMANDES DE BASES : Répertoires

Commande mkdir

- La commande mkdir (make directory) permet de créer un répertoire s'il n'existe pas.
- On peut aussi utiliser l'option -p pour créer un répertoire même si son répertoire parent n'existe pas.
- Si l'un des répertoires intermedaires n'existe pas, la commande mkdir retourne un code d'erreur (exit status) sans créer le status (sauf si l'option –p est spécifiée)
- mkdir [-p] nouveau_répertoire

Exemple: Si on veut créer le répertoire /tmp/afpa/tsrit mais le répertoire tmp/afpa/ n'existe pas, on devrait normalement faire ceci:

```
Topper Harley@debian :~$ mkdir /tmp
Topper Harley@debian :~$ mkdir /tmp/afpa/
Topper Harley@debian :~$ mkdir /tmp/afpa/tsrit
```

Si on essaie de le créer, on reçoit le message suivant:

mkdir: cannot create directory '/tmp/afpa/tsrit': No such file or directory.

Nous devrons faire:~$ mkdir -p /tmp/afpa/tsrit

COMMANDES DE BASES : Répertoires

Commande rmdir

- La commande rmdir (remove directory) permet de supprimer le répertoire spécifié sur la ligne de commande (répertoire). Si il existe des fichiers ou des sous répertoires, la commande retournera un code d'erreur (exit status). (Pour supprimer un répertoire, il faut qu'il soit vide)
- rmdir [-p] [-s] répertoire
- -p : permet de détruire tous les sous –répertoires vides.
- -s : mode silencieux (aucun affichage)
- répertoire : représente le nom du répertoire à détruire. C'est un argument obligatoire.
- **Exemple**:

#rmdir –p /tmp/afpa

#cd /tmp

tmp: does not exist

COMMANDES DE BASES : Répertoires

- **Commande rm**

La commande **rm (remove) :** Efface les fichiers ou les répertoires .

rm [Options] (fichier | répertoire)

Options :

-i Demande confirmation pour chaque fichier.

-r Destruction d'arborescence (récursive) même non vide

-f Force la destruction des fichiers pour lesquels on ne possède pas le droit d'écriture.

Exemples :

$ rm –i *.c

$ rm –rf prog

COMMANDES DE BASES : Répertoires

Commande cp (copie)

(1) cp [Options] fic1 fic2 -> Copie fic1 dans fic2

(2) cp [Options] fic1 … répertoire -> Copie un ensemble de fichiers dans un répertoire.

(3) cp –r répertoire1 répertoire2 -> Copie une arborescence dans un répertoire.

Options :

-i Demande confirmation de la destruction des fichiers cibles.

-p Conserve les droits et la date de la dernière modification des fichiers sources.

-r Copie d'une arborescence (récursive).

Exemples :

- $ cp exo.c exercice.c (copie exo.c dans exercice.c)
- $ cp cornichon ../cornichonvert (copie cornichon dans le répertoire de niveau supérieur sous le nom de cornichonvert)
- $ cp Lait Jus Biere /tmp (copie les fichiers Lait, Jus, Biere dans le répertoire /tmp)
- $ cp exercice*.* ~ (copie tous les fichiers exercice dans /home/toto par exemple)

Commande mv (déplacement ou renommage)

mv [Options] fichier1 fichier2-> Renomme fichier1 en fichier2.

mv [Options] fichier … répertoire->Transfère les fichiers dans le répertoire indiqué.

Exemple :

$ mv exo.c exercice.c

COMMANDES DE BASES

- **Commande cat**

La commande **cat (catenate « concaténer »: joindre des fichiers séquentiellement) :** Affiche des fichiers et /ou concatène les fichiers spécifiés sur la ligne de commande.

cat [fichier...]

Exemple

```
~$ cat Fruits
banane
Cerise
Orange
```

Commandes de consultation de fichiers

- cat nom_fichier : afficher le contenu d'un fichier texte à l'écran (sans pouvoir arrêter le défilement)
- more nom_fichier : afficher le contenu d'un fichier texte à l'écran en arrêtant le défilement à chaque page (espace pour descendre d'une page, entrée pour descendre d'une ligne, q pour quitter)
- less nom_fichier : pareil que more, mais dispose d'un peu plus d'options (entre autre peut revenir en arrière : b pour revenir d'une page, y pour revenir d'une ligne, il est aussi possible de se déplacer vers le haut ou vers le bas avec les flèches de direction)

Commandes de consultation de fichiers

- **Commandes head / tail**

La commande **head** : Affiche les n premières lignes d'un fichier, alors qu **tail** affiche les dernières lignes d'un fichier.
Si n n'est pas précisé, il prend par défaut la valeur 10.

head [-n] [fichier...]

tail [-n] [fichier...]

- **-n :**affiche les n dernières lignes

COMMANDES DE CONSULTATIONS

- **Commande Sort**

La commande **sort** **:** Trie les lignes du fichier par ordre alphabétique.

sort [fichier...]

Mais les options suivantes modifient les critères:

- **sort –r** Fichier : (« reverse »:inverse) trie les lignes du fichier en ordre inverse.
- **sort –n** Fichier : Effectue un trie numérique.

COMMANDES DE CONSULTATIONS

- **Commande touch**

La commande **touch** **:** est utilisée pour changer les dates d'accès et de modifications d'un fichier (d'où le « on touche »le fichier pour faire croire à l'ordinateur qu'on vient de le modifier alors qu'on n'a rien changé.