



LINUX

PRISE EN MAIN DE LINUX
(COMMANDES DE BASE)

Titre Professionnel : Développeur Web et Web
Mobile (niveau III)



GESTION DE VERSIONS

Version	Date	Rédacteur	Description
1.0	16/10/2023	Nadjet BENBAREK	CREATION DU DOCUMENT



LE SYSTÈME D'EXPLOITATION

LE SYSTÈME D'EXPLOITATION

Le système d'exploitation permet de développer des programmes sans tenir compte de la complexité physique de la machine. Le programme utilise des fonctionnalités standardisées d'accès aux ressources matériel.

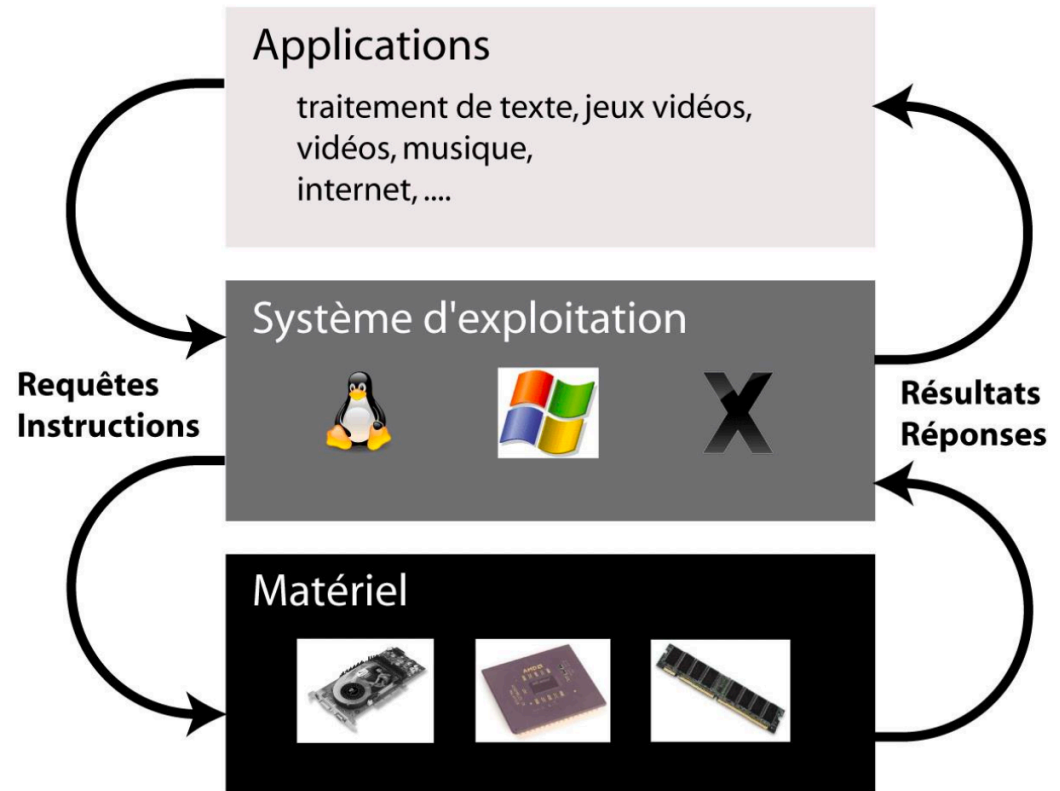
Côté Système

- coordonne l'utilisation de ces ressources (ex. : temps CPU accorde a chaque processus, allocation mémoire,. . .),
- assure la maintenance et la fiabilité du système (ex. : gestion des fichiers, de la sécurité informatique, ...)

LE SYSTÈME D'EXPLOITATION

Côté utilisateur

- facilite l'accès et l'utilisation des ressources matériel,
- propose une interface de programmation permettant d'utiliser ces matériels



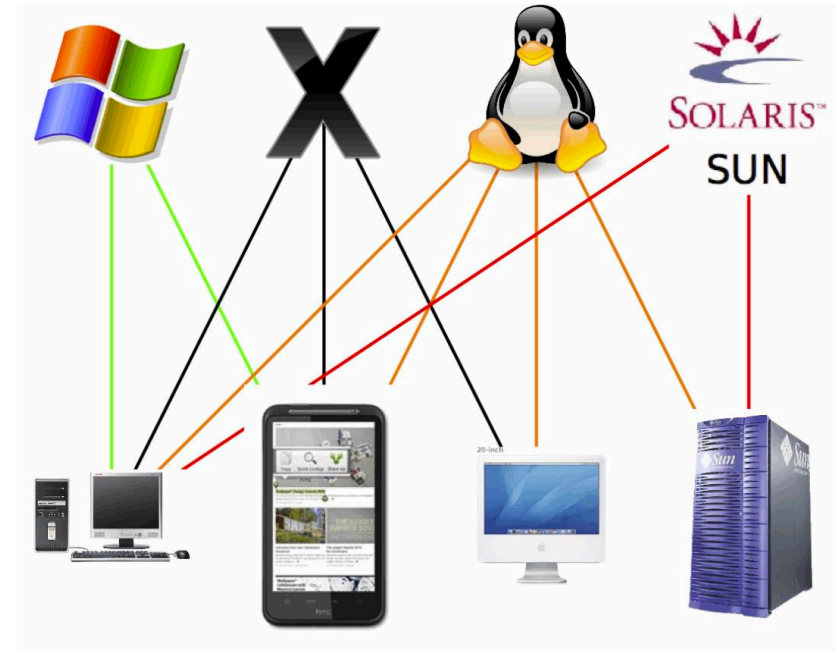
LES DIFFÉRENTS SYSTÈMES D'EXPLOITATION

Beaucoup d'OS différents existent :

Chaque architecture matériel demande un système d'exploitation adapté. Certains systèmes d'exploitation sont plus souples et prennent en charge des architectures matérielles multiples.

Deux OS se distinguent :

- Windows est le système d'exploitation le plus utilisé, et Linux est le système d'exploitation le plus souple.



GNU-LINUX – UN PEU D'HISTOIRE

Le système GNU-Linux est la rencontre d'une technologie, le noyau Linux et d'une philosophie de développement et de diffusion. C'est un système au développement collaboratif (par une communauté) qui est distribué librement et permet l'utilisation de tous les logiciels libres développés pour son architecture.

Le noyau Linux est historiquement une version libre du système UNIX développé initialement par le Finlandais Linus Torvalds à partir du début des années 1990.

Le projet GNU est celui du développement collaboratif et libre d'un système d'exploitation libre initié par Richard Stallman en 1983.

GNU-LINUX - AUJOURD'HUI

C'est un système très largement diffusé et utilisé sur lequel ont été développées plusieurs distributions (qui sont des suites logicielles qui accompagnent le noyau).

Initialement confidentiel et réservé à des spécialistes avec des interfaces rudimentaires, il est aujourd'hui toujours plus ergonomique et automatisé pour les non spécialistes, mais laisse les outils et interfaces de bas niveau disponibles au plus grand nombre.

On notera par exemple l'existence de nombreuses interfaces graphiques Bureaux (GNOME, KDE, . . .) de nombreux paquetages précompilées, de nombreux outils d'administration et de services (protocoles, . . .)

UN SYSTÈME MULTI-UTILISATEURS

Des utilisateurs et des droits

Chaque personne accédant au système est identifiée par un nom d'utilisateur (i.e. login) et un mot de passe (i.e. password).

Chaque utilisateur bénéficie de permissions : exécution de certains programmes, lecture de certaines données, écriture de fichiers dans une limite de taille et dans seulement certains répertoires.

Chaque utilisateur bénéficie d'un espace de travail réservé sur le disque. Cet espace de travail est un répertoire de l'arborescence dans lequel l'utilisateur a tous les droits : il peut y créer des sous-répertoires, y écrire des fichiers, y installer des programmes et applications. Toutes ses données et préférences personnelles y sont regroupées.

Ce répertoire est appelé "Répertoire Personnel" ou "Home Directory". Il est en général placé dans un répertoire qui s'appelle /home/ et porte le nom de l'utilisateur : /home/nom utilisateur/.

UN SYSTÈME MULTI-UTILISATEURS

Super utilisateur - Root

Certains utilisateurs ont des permissions étendues pour administrer le système et effectuer des opérations interdites à l'utilisateur normal.

L'utilisateur root a tous les droits dans le système (ex. : il peut changer les permissions de n'importe quel fichier, les noms d'utilisateur et les mots de passe, il peut installer des programmes et bibliothèques dans les répertoires système, . . .)

UTILISATION DU SHELL

Le shell correspond à une fenêtre présentant un prompt, encore appelé invite de commande. Celle-ci est paramétrable et par défaut en bash se compose comme suit :

login@machine\$

(suffixe \$ -> utilisateur normal,

suffixe # -> super-utilisateur – administrateur)

On saisit les commandes à la suite du prompt

Pour stopper la commande en cours : Ctrl-C

Pour mettre en attente la commande en cours : Ctrl-Z

Pour terminer l'entrée standard (les éventuelles paramètres donnés par l'utilisateur via le clavier) : Ctrl-D

UTILISATION DU SHELL

Le shell est personnalisable au moyen des fichiers suivants :

1. le fichier `/etc/profile`, s'il existe
2. le fichier `$HOME/.bash profile`, s'il existe
3. le fichier `$HOME/.bash login`, s'il existe
4. le fichier `$HOME/.profile`, s'il existe
5. le fichier système `/etc/bashrc`
6. le fichier caché `.bashrc`, s'il existe

LES ENTRÉES-SORTIES STANDARDS

Lors de l'exécution d'une commande, un processus est créé. Celui-ci va alors ouvrir trois flux :

`stdin` l'entrée standard, par défaut le clavier

`stdout` la sortie standard, par défaut l'écran

`stderr` la sortie d'erreur standard, par défaut l'écran.

LES REDIRECTIONS

Il est possible de rediriger les flux d'entrée-sortie au moyen d'opérateurs spécifiques :

- > redirection de la sortie standard (par exemple dans un fichier)

- < redirection de l'entrée standard

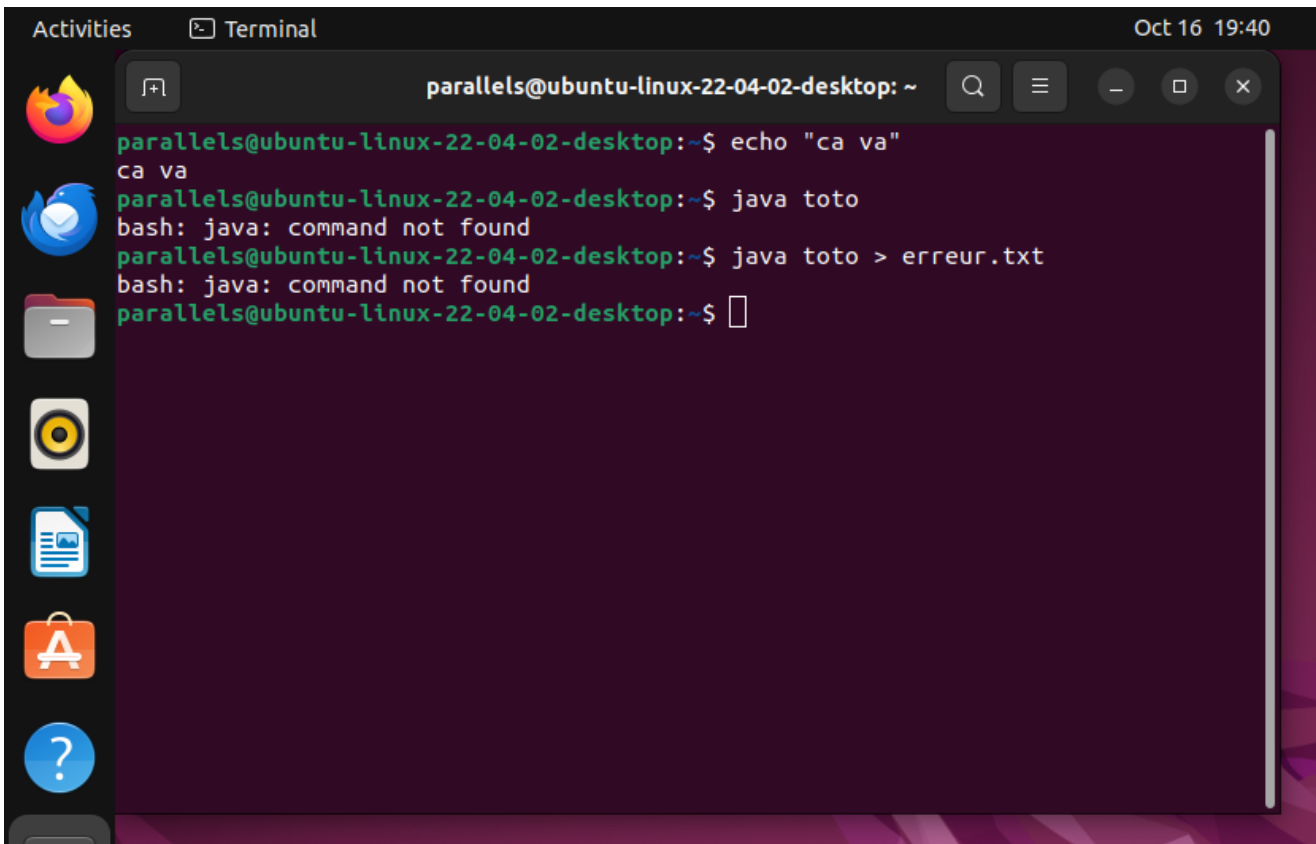
- >> redirection de la sortie standard avec concaténation

- >& redirection des sorties standard et d'erreur

- >! redirection avec écrasement de fichier

- | redirection de la sortie standard vers l'entrée standard (pipe)

EXEMPLE: LA COMMANDE ECHO



The image shows a screenshot of a Linux desktop environment with a terminal window open. The terminal window has a title bar that reads "parallels@ubuntu-linux-22-04-02-desktop: ~". The terminal output shows the following commands and their results:

```
parallels@ubuntu-linux-22-04-02-desktop:~$ echo "ca va"
ca va
parallels@ubuntu-linux-22-04-02-desktop:~$ java toto
bash: java: command not found
parallels@ubuntu-linux-22-04-02-desktop:~$ java toto > erreur.txt
bash: java: command not found
parallels@ubuntu-linux-22-04-02-desktop:~$
```

The terminal window is part of an "Activities" overview, with a sidebar on the left showing icons for the Dash, Home, and Applications menus, as well as several application icons including Firefox, Mail, Files, and the App Store. The system clock in the top right corner indicates the date and time as "Oct 16 19:40".

LE SYSTÈME DE FICHIERS

Le système de fichier correspond à une arborescence que l'on parcourt de la racine (root) vers les feuilles

La racine se note / (slash)

Il s'agit d'un répertoire contenant les sous-répertoires suivants :

/bin exécutable essentiels pour le système, directement utilisable par les utilisateurs

/boot contient les fichiers permettant à Linux de démarrer

/dev contient les points d'entrée des périphériques (=device)

/etc configuration du réseau

->contient les commandes et les fichiers nécessaires à l'administrateur du système (fichiers passwd, group, inittab, ld.so.conf, lilo.conf, ...)

LE SYSTÈME DE FICHIERS

Les sous-répertoires de la racine (suite) :

/home répertoire personnel des utilisateurs

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

/mnt contient les points de montage des partitions temporaires (cd-rom, disquette, ...), parfois nommé media

/opt contient des packages d'applications supplémentaires

/proc fichiers content des info sur la mémoire, E/S, périphérique, compatibilité pour le noyau, ...

LE SYSTÈME DE FICHIERS

Les sous-répertoires de la racine (suite) :

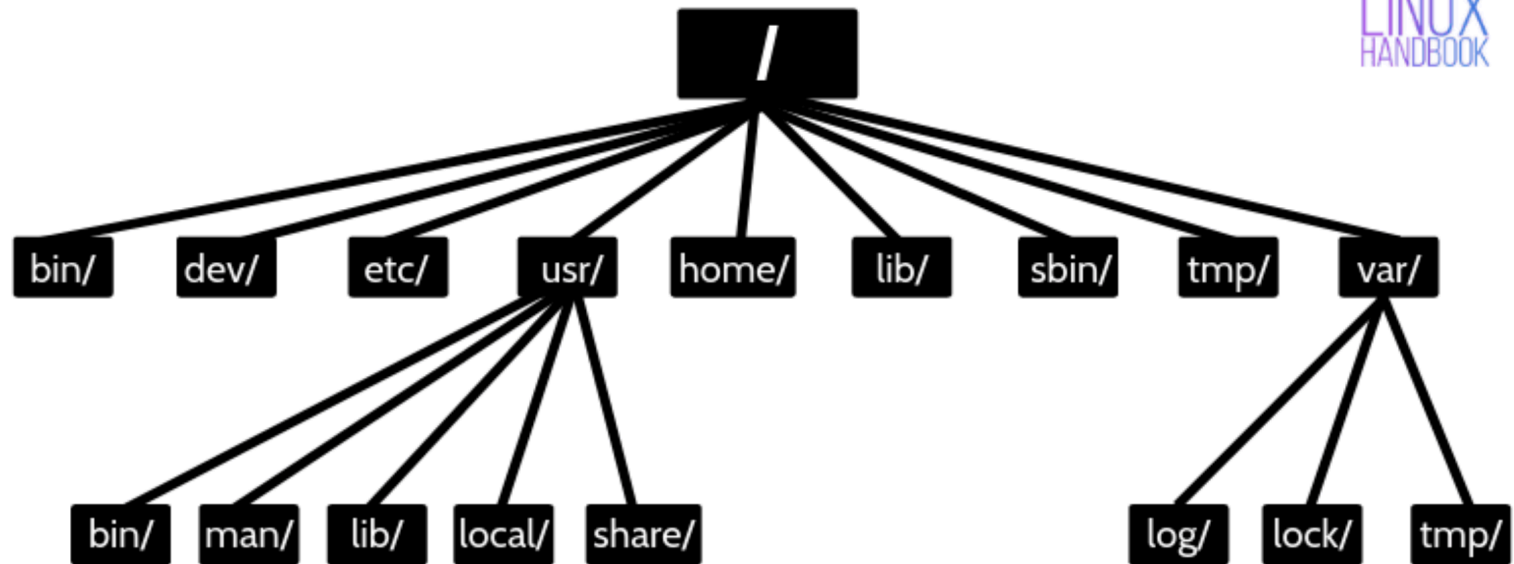
/root répertoire de l'administrateur root

/usr hiérarchie secondaire (utilisateurs)

/var contient des données variables

/tmp contient les fichiers temporaires

REPRÉSENTATION GRAPHIQUE



LINUX
HANDBOOK



LES COMMANDES FONDAMENTALES

LES COMMANDES FONDAMENTALES

Aide

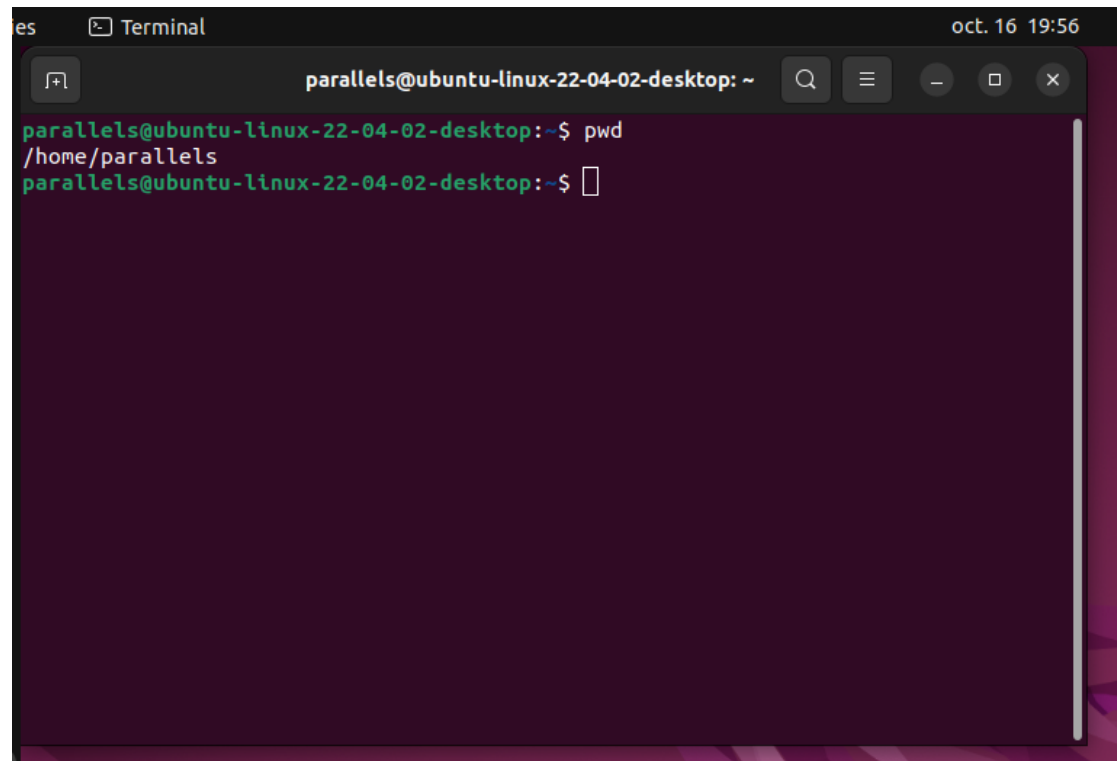
man commande

-Manuel pour les commandes

Où suis-je dans
l'arborescence ?

pwd

Exemple:

A screenshot of a terminal window titled "Terminal" with a timestamp of "oct. 16 19:56". The terminal shows the user "parallels@ubuntu-linux-22-04-02-desktop" at the prompt. The user enters the command "pwd", and the terminal outputs the current directory path "/home/parallels". The prompt then returns to the user.

```
parallels@ubuntu-linux-22-04-02-desktop:~$ pwd
/home/parallels
parallels@ubuntu-linux-22-04-02-desktop:~$
```

LES COMMANDES FONDAMENTALES

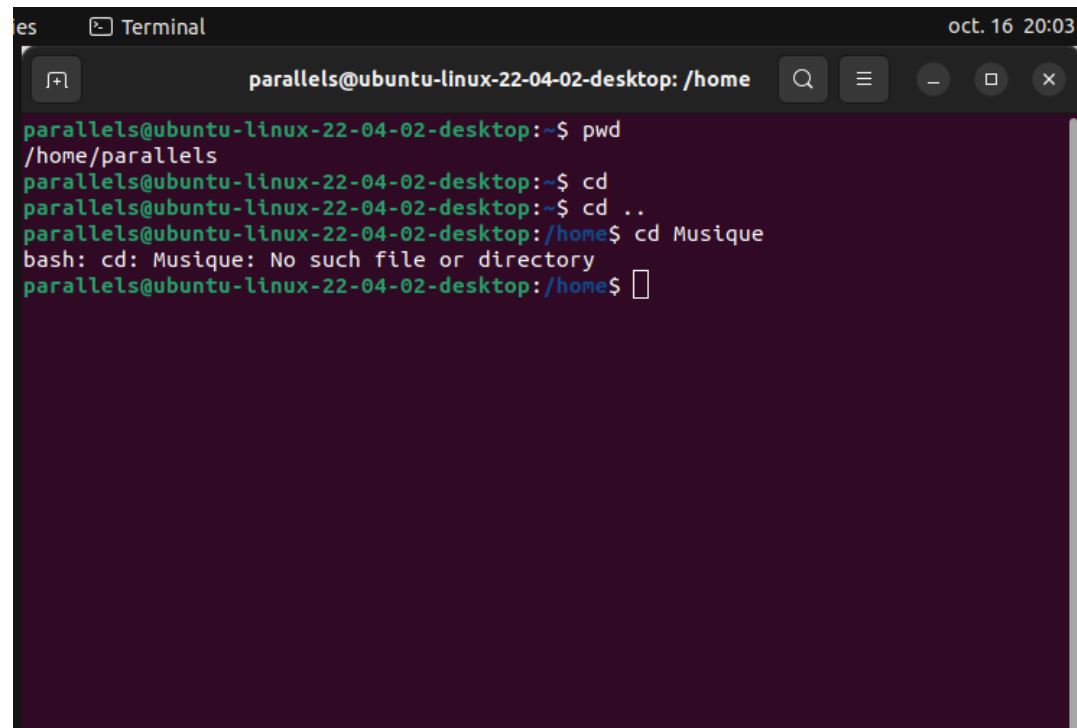
Comment se déplacer dans
l'arborescence ?

cd [chemin] -Permet de changer
de répertoire (change directory)

Alias :

- . répertoire courant
- .. répertoire parent

Exemples :

A terminal window titled "Terminal" with a dark background. The window shows a series of commands and their outputs. The user is logged in as "parallels" on an "ubuntu-linux-22-04-02-desktop" machine. The current directory is "/home". The commands and outputs are: "pwd" returns "/home/parallels"; "cd" returns no output; "cd .." returns no output; "cd Musique" returns an error message "bash: cd: Musique: No such file or directory". The prompt is always "parallels@ubuntu-linux-22-04-02-desktop:~/home\$".

```
parallels@ubuntu-linux-22-04-02-desktop:~/home$ pwd
/home/parallels
parallels@ubuntu-linux-22-04-02-desktop:~/home$ cd
parallels@ubuntu-linux-22-04-02-desktop:~/home$ cd ..
parallels@ubuntu-linux-22-04-02-desktop:~/home$ cd Musique
bash: cd: Musique: No such file or directory
parallels@ubuntu-linux-22-04-02-desktop:~/home$
```

LES COMMANDES FONDAMENTALES

Lister le contenu d'un répertoire ?

`ls [option] [chemin]`

-Liste le contenu d'un répertoire avec plus ou moins de détails

Exemples :

ls l* liste tous les fichiers commençant par l

ls -l liste tous les fichiers du répertoire courant, en donnant les attributs des fichiers (droits, taille, etc)

ls -a liste tous les fichiers du répertoire courant (y compris les fichiers cachés dont le nom commence par un ".")

man ls affiche la page de manuel de la commande ls

LES COMMANDES FONDAMENTALES

Visualiser le contenu d'un fichier ?

`cat [option] [chemin vers le fichier1, fichier2, etc]`
-affiche le contenu d'un fichier

Exemples :

`cat .bash profile` affiche le contenu du fichier caché `.bash profile`
`cat toto > tata` écrit le contenu du fichier `toto` dans un fichier nommé `tata`

Visualiser le contenu d'un fichier page à page ?

`more [fichier]`

Visualiser le contenu d'un fichier dans un flux ?

`less [fichier]`

LES COMMANDES FONDAMENTALES

Obtenir des statistiques sur le contenu d'un fichier ?

`wc [option] [chemin vers le fichier]`
-affiche le nombre de mots / lignes / caractères d'un fichier

Exemples :

`wc -l toto` affiche le nombre de lignes du fichier toto
`wc -c toto` affiche le nombre de caractères du fichier toto
`ls | wc -l` affiche le nombre de fichiers dans le répertoire courant

Editer un fichier ?

`nano [fichier]`
`emacs [fichier]`
`vim [fichier]`
`gedit [fichier]`

LES COMMANDES FONDAMENTALES

Copier un fichier ?

`cp [option] [chemin vers fichier source] [chemin vers fichier destination]`
-copie un fichier source en le renommant si le chemin du fichier destination contient un nom de fichier

Exemples :

`cp toto /tmp/` copie le fichier local toto dans /tmp(toujours nommé toto)
`cp toto /tmp/tata` copie le fichier local toto dans /tmp en le nommant tata
`cp -r projet /tmp` copie le contenu du répertoire projet dans le répertoire

LES COMMANDES FONDAMENTALES

Déplacer un fichier ?

`mv [option] [chemin vers fichier source] [chemin vers fichier destination]`
-déplace un fichier source en le renommant si le chemin du fichier destination contient un nom de fichier

Exemples :

`mv toto /tmp/` déplace le fichier local toto dans /tmp (toujours nommé toto)
`mv toto /tmp/tata` déplace le fichier local toto dans /tmp en le nommant tata
`mv -i toto /tmp` déplace le fichier toto dans /tmp en prévenant l'utilisateur s'il existe déjà un fichier tmp/toto

LES COMMANDES FONDAMENTALES

Supprimer un fichier ?

```
rm [option] [chemin vers fichier]  
-supprime un fichier
```

Exemples :

```
rm toto supprime le fichier toto  
rm -i toto supprime le fichier toto en demandant confirmation à l'utilisateur  
rm -f toto* supprime les fichiers dont le nom commence par toto, sans demander  
confirmation à l'utilisateur  
rm -r projet efface récursivement le contenu du répertoire projet
```


LES COMMANDES FONDAMENTALES

Créer / supprimer un répertoire ?

mkdir [chemin vers répertoire]
rmdir [chemin vers répertoire]
-crée / supprime un répertoire vide

Exemples :

mkdir toto crée le répertoire toto
rmdir toto supprime le répertoire vide toto
rmdir projet rmdir: projet/: Directory not empty

LES COMMANDES FONDAMENTALES

Retrouver un fichier ?

`find [options]`

-effectue une recherche à partir des informations données en option

Exemples :

`find . -name toto` cherche, dans le répertoire courant et ses sous-répertoires, un fichier nommé toto

`find /tmp/ -type d` cherche tous les sous-répertoires du répertoire /tmp

LES COMMANDES FONDAMENTALES

Retrouver un fichier exécutable ?

which commande

-effectue une recherche dans la liste des exécutables de la commande donnée

Exemples :

which echo /bin/echo

Connaître l'espace occupé par un répertoire / disque ?

du [option] fichier

-donne la taille en octets d'un fichier

df [option]

-donne la taille des données présentes sur chaque disque

Exemples :

du -sh projet 4.0K projet/

LES COMMANDES FONDAMENTALES

Rechercher un motif dans un fichier ?

`grep [options] expression régulière fichier1 ...`

-effectue une recherche à partir d'un motif fourni dans une expression régulière donnée

Exemples :

`grep "grep" /home/nadjet/cours/*` cherche, dans les fichiers du répertoire cours, des fichiers contenant le motif « grep »

`grep -n "grep" /home/nadjet/cours/*` idem, mais en affichant le numéro de ligne

`grep -c "grep" /home/nadjet/cours/*` idem, mais en donnant le nombre d'occurrences du motif

LES COMMANDES FONDAMENTALES

Compresser/décompresser un fichier ?

gzip fichier

-compresse un fichier au format .gz (algorithme deflate)

gunzip fichier

-décompresse un fichier au format .gz

Exemples :

gzip toto.txt toto.txt.gz

gunzip toto.txt.gz toto.txt

LES COMMANDES FONDAMENTALES

Créer/extraire une archive ?

`tar cf projet.tar projet/*`

-crée une archive contenant le contenu du répertoire projet et nommée projet.tar

`tar xf projet.tar`

-extraît le contenu de l'archive nommée projet.tar

`tar zcf projet.tar projet/*`

-crée et compresse une archive contenant le contenu du répertoire projet et nommée projet.tar.gz

`tar zxf projet.tar.gz`

-extraît le contenu de l'archive compressée nommée projet.tar.gz

LES COMMANDES FONDAMENTALES

Créer un lien vers un fichier ?

In [options] fichier source lien

-création d'un lien. un lien est un type spécial de fichier qui permet à plusieurs noms de fichiers de faire référence au même fichier sur le disque.

NB: lien "dur" vs lien symbolique

Exemples :

In /home/nadjet/cours.pdf /home/nadjet/projet/cours.pdf

-le fichier cours.pdf du répertoire nadjet du répertoire projet est un lien vers le fichier cours.pdf du répertoire nadjet

In -s /home/nadjet/cours.pdf /home/nadjet/projet/cours.pdf

-idem avec un lien symbolique

LES COMMANDES FONDAMENTALES

Connaître les ressources prises par une commande ?

`ps [options]`

-donne des informations sur les processus en cours

Exemple :

`ps ux`

Connaître l'activité du système ?

`top`

-donne des informations sur l'activité du système (ressources occupées, etc)

LES COMMANDES FONDAMENTALES

Interrompre un processus ?

`kill [option] PID`

-envoie un signal au processus identifié par le nombre PID
(si l'option est -9, ce signal provoque l'interruption de la commande)

Trier le contenu d'un fichier ?

`sort fichier`

Savoir qui est connecté au système ?

`who`



LES COMMANDES D'ADMINISTRATION

LES COMMANDES D'ADMINISTRATION

Gérer les droits d'un fichier ?

`chmod [options] droits fichier1, fichier2, ...`
-change les droits d'un fichier

Les droits sont définis comme suit :

u droits de l'utilisateur (user)

g droits des utilisateurs du groupe (group)

a droits de tous les utilisateurs (all)

+r droit en lecture accordé

-r droit en lecture retiré

+w droit en écriture accordé

+x droit d'exécution accordé

Exemples :

chmod a+r toto.txt autorise l'accès en lecture du fichier toto.txt à tout le monde

chmod 444 toto.txt idem

LES COMMANDES D'ADMINISTRATION

Changer le propriétaire et groupe d'un fichier ?

`chown [options] utilisateur.groupe fichier`

-change le propriétaire d'un fichier

NB: nécessite d'être administrateur (super-user)

Exemple :

`sudo chown -R saman.L1 projet/`

-définit l'utilisateur saman et le groupe L1 au répertoire projet et à tous ses fichiers

Ajouter un utilisateur ?

`useradd [options] login`

Changer de mot de passe ?

`passwd [options] login`



LES VARIABLES D'ENVIRONNEMENT

LES COMMANDES D'ADMINISTRATION

Variables permettant de paramétrer le fonctionnement du système (langue utilisée, chemins vers les fichiers exécutables, chemin vers les librairies, etc)

Variables principales :

SHELL interpréteur de commande utilisé HOME chemin du répertoire de l'utilisateur

PATH chemin des exécutables

Lire une variable d'environnement : `echo $HOME` Définir une variable d'environnement (bash) :

```
export PATH=$PATH:/home/nadjet/myexec
```

De nombreuses variables d'environnement sont définies dans le fichier `$HOME/.bash profile` (bash)