

Unix/Linux



PLAN

I. Généralités

Qu'est-ce qu'un ordinateur ?

Les composants principaux et les principes de fonctionnement d'un ordinateur

II. Le système d'exploitation

III. Le système Linux

IV. Fichiers et répertoires

Définition

Définition

"Un ordinateur est une machine électronique programmable capable de réaliser des calculs logiques sur des nombres binaires.

C'est une machine

Le fonctionnement d'un ordinateur est basé sur une architecture matérielle (processeur, support de stockage, interfaces utilisateurs, connexion, . . .) dont le fonctionnement est soumis aux lois de la physique.

C'est une machine programmable

Cette machine est capable de remplir des tâches différentes selon les instructions qui lui sont adressées. Ces instructions, rédigées sous forme de programmes par les informaticiens, sont traitées en séquence par le matériel de l'ordinateur.

Interaction avec le matériel

Heureusement, la plupart du temps, l'informaticien n'a pas à interagir directement avec le matériel. Pour traiter avec les composants, tous les ordinateurs modernes disposent d'une couche logicielle appelée Système d'Exploitation. Cette couche est en charge de faire la passerelle entre l'informaticien, ses outils, les programmes qu'il développe et, les composants et leur fonctionnement.

Points communs et différences

= Matériel

Des capacités de calcul

CPU et/ou GPU

De la mémoire

RAM, Disque dur, . . .

= Logiciels

Pour dialoguer avec le matériel

Système d'exploitation, Firmware

Pour accomplir ses tâches

logiciels, programmes, . . .

Différent Périphériques

Interfaces

Connexions réseau, écrans, claviers, . . .

Les différents systèmes d'exploitation



Linux

Non propriétaire : Gratuit le plus souvent
Ouvert : sources disponibles Flexible : sources modifiables
Puissant : Programmable Communauté active : entraide des utilisateurs
Plus complexe : pour les informaticiens (interface de programmation optimisées)



Windows©

Propriétaire : Payant
Sources non disponibles
Sources non modifiables
Plus difficilement programmable
Communauté active : nombreux utilisateurs
Plus adapté : pour les utilisateurs (interfaces d'utilisation optimisées)

Les différents systèmes d'exploitation

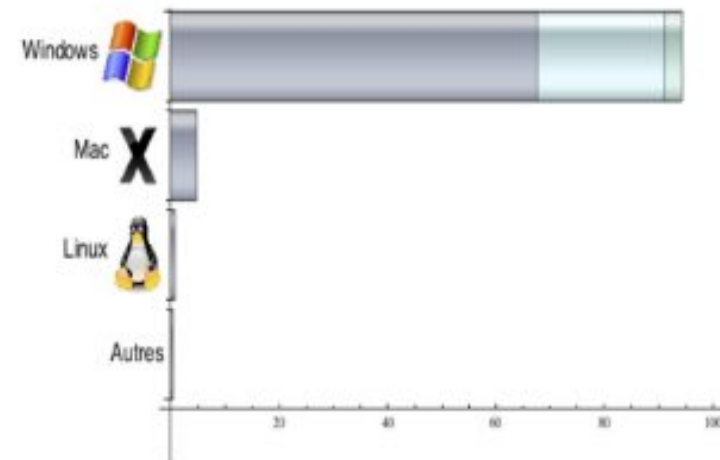
Beaucoup d'OS différents existent :

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.



”

Les différents systèmes d'exploitation



Linux

Non propriétaire : Gratuit le plus souvent
Ouvert : sources disponibles Flexible : sources modifiables
Puissant : Programmable Communauté active : entraide des utilisateurs
Plus complexe : pour les informaticiens (interface de programmation optimisées)

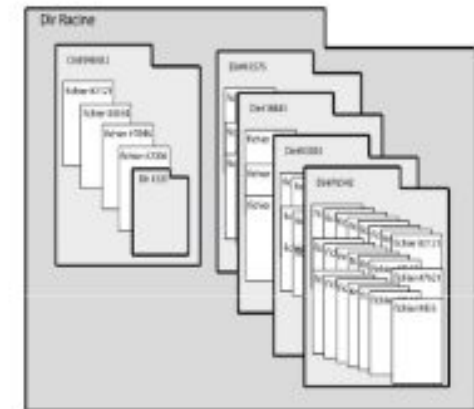


Windows©

Propriétaire : Payant
Sources non disponibles
Sources non modifiables
Plus difficilement programmable
Communauté active : nombreux utilisateurs
Plus adapté : pour les utilisateurs (interfaces d'utilisation optimisées)

Organisation des données enregistrées

- De très nombreux fichiers et répertoires
- Le nombre de fichiers enregistrés sur un disque dur peut aisément dépasser 100.000 fichiers,
- Chaque fichier est identifié par un nom,
- Les fichiers sont regroupés dans des répertoires et sous-répertoires.
- Chaque répertoire est identifié par un nom.

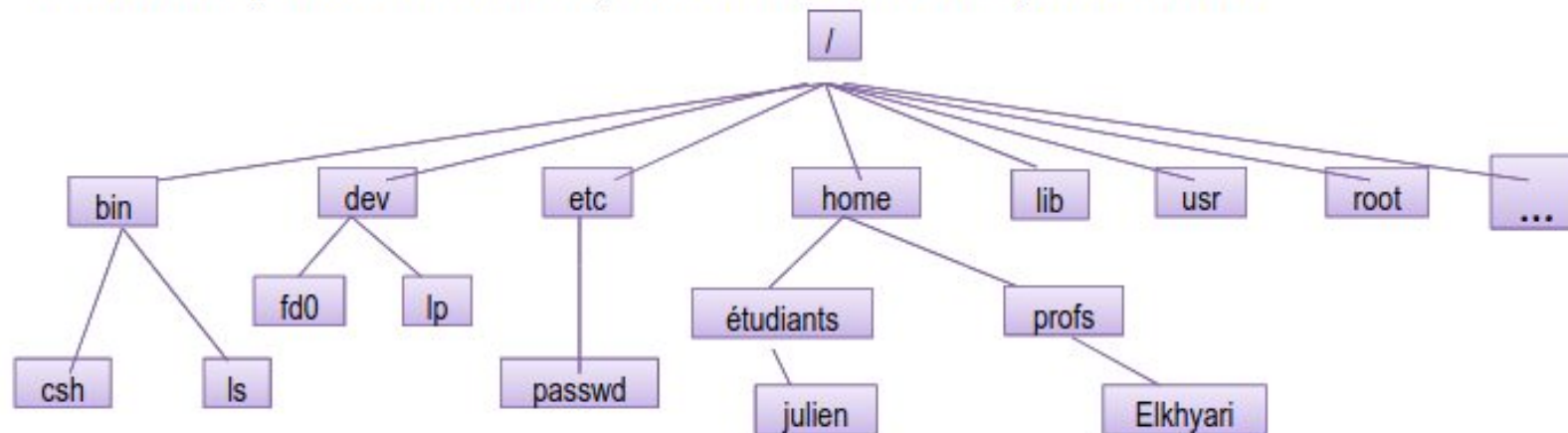


Une organisation en arborescence

- Cette organisation arborescente permet de faciliter la recherche d'un fichier,
- Les fichiers sont regroupés par application, par thème, par format, par fonction, . . .

La hiérarchie des répertoires

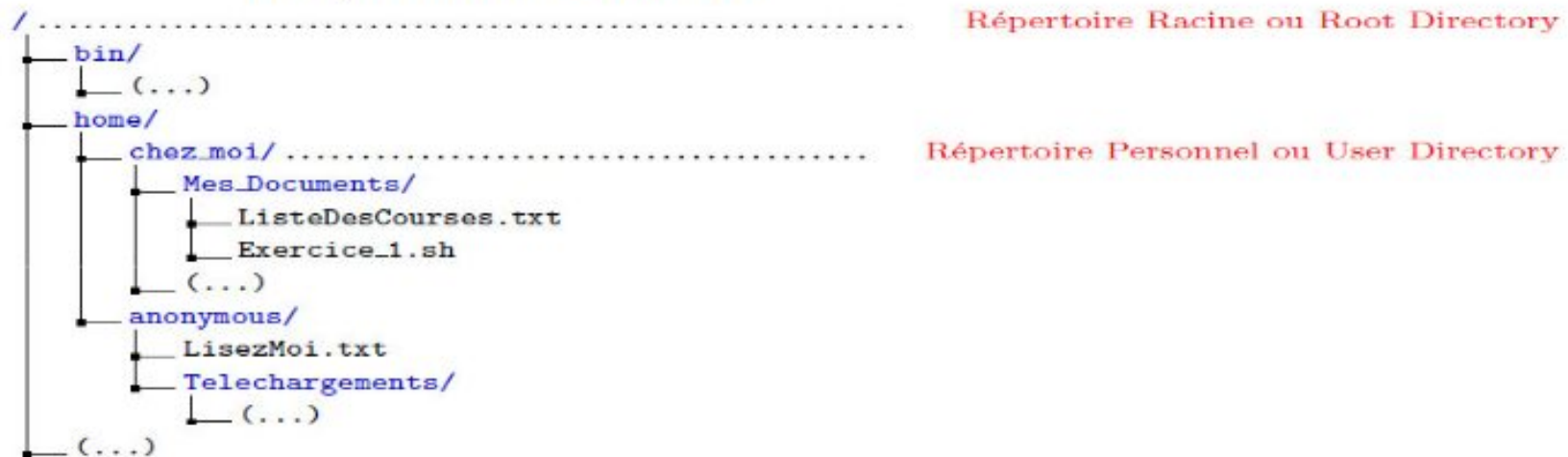
- Linux définit un système de fichiers hiérarchique avec un certain nombre de répertoires standards



- `/root` est le répertoire d'accueil du super-utilisateur (administrateur)
- `/bin` contient généralement les programmes utiles au démarrage
- `/etc` contient les fichiers de configurations
- `/dev` contient les fichiers relatifs aux devices (périphériques)
- `/home` contient les répertoires des utilisateurs
- `/lib` contient les bibliothèques du système
- `/usr` contient les programmes ajoutés au système

L'Organisation Arborescente

Exemple d'arborescence Linux



Les répertoires importants

Le Répertoire Racine (Root directory) contient tous les répertoires et fichiers accessibles depuis le système.
Le Répertoire Personnel (User Directory ou Home Directory) est le répertoire dans lequel l'utilisateur peut faire ce qu'il veut (écrire, modifier, supprimer, installer ...).

La notion de Chemin

Le chemin définit un nom unique

- Deux fichiers ou répertoires ne peuvent pas porter le même nom si ils sont dans un même répertoire.
- Les noms des fichiers et répertoires différencient les caractères Majuscules

EXEMPLES DE CHEMINS ABSOLUS

/	Un chemin absolu part de la racine /
├─ home/	/home/
│ └─ chez_moi/	/home/chez_moi/
│ │ └─ Etoiles/	/home/chez_moi/Etoiles/
│ │ │ └─ SOLEIL.jpg	/home/chez_moi/Etoiles/SOLEIL.jpg
│ │ │ └─ Soleil.jpg	/home/chez_moi/Etoiles/Soleil.jpg
│ │ └─ Systeme_Solaire/	/home/chez_moi/Systeme_Solaire/
│ │ └─ SOLEIL.jpg	/home/chez_moi/Systeme_Solaire/SOLEIL.jpg

Syntaxe d'un chemin absolu

Le chemin absolu d'un fichier ou d'un répertoire est unique. Il donne la liste des répertoires et sous-répertoires en partant de la racine / (la référence absolue de l'arborescence) jusqu'à la cible.

Notation Spéciales

Les chemins des répertoires de référence

Répertoire	Notation
Répertoire Racine	/
Répertoire Personnel	~

Répertoire	Notation
Répertoire Courant	.
Répertoire Parent	..

Remarques

La notation ~ correspond a un chemin absolu. Elle est remplacée lors d'une évaluation par le chemin absolu du répertoire personnel de l'utilisateur.

Exemple de chemins valides pointant le fichier cible

```
/..... Répertoire Racine
├── home/
│   ├── chez_moi/ ..... Répertoire Personnel
│   │   ├── Etoiles/.... Répertoire Courant
│   │   │   └── Soleil.jpg..... Fichier cible
```

Chemins Absolus

```
/home/chez_moi/Etoiles/Soleil.jpg
~/Etoiles/Soleil.jpg
/home/chez_moi/../../chez_moi/Etoile/Soleil.jpg
/home/chez_moi/../../home/chez_moi/Etoile/Soleil.jpg
```

Chemins Relatifs

```
Soleil.jpg
../Etoile/Soleil.jpg
../../chez_moi/Etoile/Soleil.jpg
```

”

Conventions

Commandes, options, paramètres

- **Commande** c'est le nom d'un programme qui exécute une action.
- **Options** ce sont des paramètres optionnels. Ils peuvent être négligé. L'ajout d'options modifie le comportement de la commande (le résultat). Les options sont encadrées par les caractères < options >.
- **Paramètres** ce sont des arguments que la commande évalue.

Sources et Cible

- **Source** c'est un fichier ou un répertoire utilise en entrée d'une commande,
- **Cible** c'est un fichier ou un répertoire utilise en sortie d'une commande.

Manipulation de l'arborescence en ligne de commande

Principales commandes

Commande	Fonction principale
pwd	Afficher le nom du répertoire courant
ls	Afficher le contenu d'un répertoire
cd	Changer de répertoire courant
mkdir	Créer un répertoire
rm	Supprimer fichier(s) ou répertoire(s)
cp	Copier fichier(s) ou répertoire(s)
mv	Déplacer/Renommer fichier(s) ou répertoire(s)

pwd

Syntaxe

pwd

Description

Affiche le nom du répertoire courant.

EXEMPLE D'UTILISATION:

```
/ ..... Répertoire Racine
├── home/
│   ├── chez_moi/ ..... Répertoire Courant
│   └── Etoiles/
```

```
/ ..... Répertoire Racine
├── home/
│   ├── chez_moi/ ..... Répertoire Personnel
│   └── Etoiles/ ..... Répertoire Courant
```

Cliqu

“

Manipulation de l'arborescence en ligne de commande

Ls

Syntaxe

ls <source>

Description

- Affiche le contenu d'un répertoire.
- Par défaut si aucune source n'est indiquée, la commande affiche le contenu du répertoire courant.

Exemple d'utilisation:

```
/ ..... Répertoire Racine
├── home/ ..... Répertoire Courant
│   ├── chez_moi/
│   │   ├── astronomie.txt
│   │   └── Etoiles/
```

```
[ login@localhost /home/ ] ls
chez_moi/
```

```
[ login@localhost /home/ ] ls chez_moi/
Etoiles/ astronomie.txt
```

”

Manipulation de l'arborescence en ligne de commande

ls(bis)

Syntaxe

`ls -a <source>`

Description

- Affiche le contenu d'un répertoire y compris les fichiers et répertoires caches.
- Les fichiers et répertoires caches ont un nom dont le premier caractère est un point.
- Les fichiers et répertoires caches sont utilisés par le système ou certaines applications.

Exemple d'utilisation:

chez_moi/.... Rép. Courant

```
├── ./ssh/
│   ├── id_rsa
│   ├── id_rsa.pub
│   └── known_hosts
├── .bashrc
├── astronomie.txt
├── Etoiles/
└── soleil.jpg
```

Sans option -a

```
[ login@localhost ~ ] ls
astronomie.txt
Etoiles/
[ login@localhost ~ ] █
```

Avec option -a

```
[ login@localhost ~ ] ls -a
.
..
.ssh/
.bashrc
astronomie.txt
Etoiles/
[ login@localhost ~ ] █
```

”

Manipulation de l'arborescence en ligne de commande

Cd

Syntaxe

`cd <cible>`

Description

- Change le répertoire courant (permet de naviguer dans l'arborescence).
- Si le chemin du répertoire cible est omit, le répertoire courant redevient par défaut le répertoire personnel.

Exemple d'utilisation:



Commande #1 :

```
[ login@localhost /home ] cd
[ login@localhost ~ ] █
```

Commande #2 :

```
[ login@localhost /home ] cd chez_moi/Etoile
[ login@localhost ~/Etoile ] █
```

”

Manipulation de l'arborescence en ligne de commande

mkdir

Syntaxe

`mkdir chemin <chemin 2 ...>`

Description

- Création d'un ou de plusieurs répertoires aux endroits spécifiés par les chemins.
- Si le chemin est occupé par un fichier ou un répertoire, il y a un message d'erreur.

Exemple d'utilisation:

```
chez_moi/..... Répertoire Courant
├── astronomie.txt
├── Systeme_Solaire/..... Création Commande #1
├── Etoiles/
│   ├── Rouges/..... Création Commande #2
│   └── Bleues/..... Création Commande #3
└── Galaxies/..... Création Commande #3
```

Commande #1 :

```
[ login@localhost ~ ] mkdir Systeme_Solaire
```

Commande #2 :

```
[ login@localhost ~ ] mkdir Etoiles/Rouges
```

Commande #3 :

```
[ login@localhost ~ ] mkdir Galaxies Etoiles/Bleues
```

Cliquez

“

”

Manipulation de l'arborescence en ligne de commande

rm

Syntaxe

`rm chemin <chemin 2 ...>`

Description

- La commande supprime le fichier pointé par le(s) chemin(s).
- Si le chemin pointe sur un répertoire, la commande affiche un message d'erreur.

Exemple d'utilisation:

```
chez_moi/..... Répertoire Courant
├── astronomie.txt ..... Supprimé par la Commande #1
├── Etoiles/
│   ├── soleil.jpg ..... Supprimé par la Commande #2
│   └── aldebaran.gif ..... Supprimé par la Commande #2
```

Commande #1 : `[login@localhost ~] rm astronomie.txt`

Commande #2 : `[login@localhost ~] rm aldebaran.gif Etoiles/soleil.jpg`

Manipulation de l'arborescence en ligne de commande

rm(bis)

Syntaxe

`rm -r chemin <chemin 2 ...>`

Description

L'option -r (Récursif) permet de supprimer un répertoire et tout son contenu.

Exemple d'utilisation:

```
chez_moi/..... Répertoire Courant
├── astronomie.txt
├── Etoiles/..... Supprimé par la Commande #1
│   ├── soleil.jpg..... Supprimé par la Commande #1
│   └── Galaxie/..... Supprimé par la Commande #1
│       └── Andromede.pdf..... Supprimé par la Commande #1
└── aldebaran.gif
```

Commande #1 : `[login@localhost ~] rm -r Etoiles`

Manipulation de l'arborescence en ligne de commande

cp

Syntaxe

cp source cible

Description

- Copie le fichier source vers la cible.
- La source doit être un fichier ordinaire (pas un répertoire),
- Si la source est un répertoire la commande produit un message d'erreur.
- Si la cible :
 - * est le chemin d'un répertoire existant, le fichier sera copié dans ce répertoire et conservera son nom,
 - * ne correspond pas à un répertoire existant, le fichier sera copié avec le nom cible.

Exemple d'utilisation

```
chez_moi/..... Répertoire Courant
├── astronomie.txt ..... Fichier Source Commande #1
├── Etoiles/ ..... Répertoire Cible Commande #1
│   └── astronomie.txt ..... Copié/Créé par la Commande #1
└── cv.pdf
```

Commande #1 :

```
[ login@localhost ~ ] cp astronomie.txt Etoiles
```

”

21

10/01/23

Manipulation de l'arborescence en ligne de commande

cp(bis)

Syntaxe

`cp source <source 2 ...> cible`

Description:

- Copie plusieurs fichiers sources vers la cible.
- Les sources doivent être des fichiers ordinaires, et la cible un répertoire.

Exemple d'utilisation:

chez_moi/.....	Répertoire Courant
├─ cv.pdf	Fichier Source Commande #2
├─ motivations.pdf	Fichier Source Commande #2
└─ Candidature/	Répertoire Cible Commande #2
├─ cv.pdf	Copié/Créé par la Commande #2
└─ motivations.pdf	Copié/Créé par la Commande #2

Commande #2 : `[login@localhost ~] cp cv.pdf motivations.pdf Candidature`

Manipulation de l'arborescence en ligne de commande

cp(ter)

Syntaxe

`cp -r source <source 2 ...> cible`

Description

- L'option -r (Récursif) permet de copier un répertoire et son contenu si il apparait dans le(s) source(s).

Exemple d'utilisation:

```
chez_moi/ ..... Répertoire Courant
├── astronomie.txt
├── Galaxie/ ..... Fichier Source Commande #3
│   ├── Andromede.pdf
│   └── Etoiles/ ..... Répertoire Cible #3
│       ├── soleil.jpg
│       ├── Galaxie/ ..... Copié/Créé par la Commande #3
│       │   └── Andromede.pdf ..... Copié/Créé par la Commande #3
└── aldebaran.gif
```

Commande #3 : `[login@localhost ~] cp -r Galaxies Etoiles`

”

Manipulation de l'arborescence en ligne de commande

mv

Syntaxe

mv source cible

Description

- Déplace/Renomme un fichier ou répertoire.
- modifie le chemin d'accès à la source qui devient le chemin cible.
- Le chemin source disparaît et le chemin cible est créé.
- Le fichier ou répertoire pointe reste le même.
- La cible doit être un chemin non occupé ou un répertoire.

Exemple d'utilisation: Renommer un fichier

État Initial de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── AstroNomIe.TXT..... Fichier Source
```

État Final de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── astronomie.txt..... Fichier Renommé
```

```
[ login@localhost ~ ] mv AstroNomIe.TXT astronomie.txt
```

Manipulation de l'arborescence en ligne de commande

mv(bis)

Exemple d'utilisation: Déplacer un Répertoire

État Initial de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── astronomie.txt..... Fichier Source
└── Etoiles/..... Répertoire Cible
```

État Final de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── Etoiles/..... Répertoire Cible
└── astronomie.txt..... Fichier Déplacé
```

```
[ login@localhost ~ ] mv astronomie.txt Etoiles
```

Exemple d'utilisation: Renommer un Répertoire

État Initial de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── Etoiles/..... Répertoire Source
└── astronomie.txt
```

État Final de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── Relativite/..... Répertoire Renommé
└── astronomie.txt
```

```
[ login@localhost ~ ] mv Etoiles Relativite
```

Manipulation de l'arborescence en ligne de commande

mv(ter)

Exemple d'utilisation:

État Initial de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── astronomie.txt..... Fichier Source
├── relativite.pdf..... Fichier Source
└── Etoiles/..... Répertoire Cible
```

État Final de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── Etoiles/..... Répertoire Cible
│   ├── astronomie.txt..... Fichier Déplacé
│   └── relativite.pdf..... Fichier Déplacé
```

```
[ login@localhost ~ ] mv astronomie.txt relativite.pdf Etoiles
```

Exemple d'utilisation:

État Initial de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── relativite.pdf..... Fichier Source
├── Etoiles/..... Répertoire Source
│   └── astronomie.txt
└── Espace/..... Répertoire Cible
```

État Final de l'arborescence :

```
chez_moi/..... Répertoire Courant
├── Espace/..... Répertoire Cible
│   ├── relativite.pdf..... Fichier Déplacé
│   ├── Etoiles/..... Répertoire Déplacé
│   └── astronomie.txt..... Fichier Déplacé
```

```
[ login@localhost ~ ] mv relativite.pdf Etoiles Espace
```