

Définition

Un **système d'exploitation** (en abrégé **OS**, de l'anglais *Operating System*) est un programme (ou ensemble de programme) permettant de

Définition

Un **système d'exploitation** (en abrégé **OS**, de l'anglais *Operating System*) est un programme (ou ensemble de programme) permettant de gérer les ressources de l'ordinateur (mémoire, fichier, périphériques, ...) sur lequel il s'exécute.

C1 Système d'exploitation

Définition

Un **système d'exploitation** (en abrégé **OS**, de l'anglais *Operating System*) est un programme (ou ensemble de programme) permettant de gérer les ressources de l'ordinateur (mémoire, fichier, périphériques, ...) sur lequel il s'exécute.

Exemples

Les systèmes d'exploitation les plus répandus à l'heure actuelle sont :

C1 Système d'exploitation

Définition

Un **système d'exploitation** (en abrégé **OS**, de l'anglais *Operating System*) est un programme (ou ensemble de programme) permettant de gérer les ressources de l'ordinateur (mémoire, fichier, périphériques, ...) sur lequel il s'exécute.

Exemples

Les systèmes d'exploitation les plus répandus à l'heure actuelle sont :

- Windows (différentes versions)

C1 Système d'exploitation

Définition

Un **système d'exploitation** (en abrégé **OS**, de l'anglais *Operating System*) est un programme (ou ensemble de programme) permettant de gérer les ressources de l'ordinateur (mémoire, fichier, périphériques, ...) sur lequel il s'exécute.

Exemples

Les systèmes d'exploitation les plus répandus à l'heure actuelle sont :

-  Windows (différentes versions)
-  GNU/Linux (plusieurs centaines de distribution différentes, parmi les plus connus : ubuntu, fedora, archlinux)

C1 Système d'exploitation

Définition

Un **système d'exploitation** (en abrégé **OS**, de l'anglais *Operating System*) est un programme (ou ensemble de programme) permettant de gérer les ressources de l'ordinateur (mémoire, fichier, périphériques, ...) sur lequel il s'exécute.

Exemples

Les systèmes d'exploitation les plus répandus à l'heure actuelle sont :

-  Windows (différentes versions)
-  GNU/Linux (plusieurs centaines de distribution différentes, parmi les plus connus : ubuntu, fedora, archlinux)
-  Android (smartphone)

C1 Système d'exploitation

Définition

Un **système d'exploitation** (en abrégé **OS**, de l'anglais *Operating System*) est un programme (ou ensemble de programme) permettant de gérer les ressources de l'ordinateur (mémoire, fichier, périphériques, ...) sur lequel il s'exécute.

Exemples

Les systèmes d'exploitation les plus répandus à l'heure actuelle sont :

-  Windows (différentes versions)
-  GNU/Linux (plusieurs centaines de distribution différentes, parmi les plus connus : ubuntu, fedora, archlinux)
-  Android (smartphone)
-  MacOS (ordinateur) et iOS (smartphone)

La place du système d'exploitation

La place du système d'exploitation

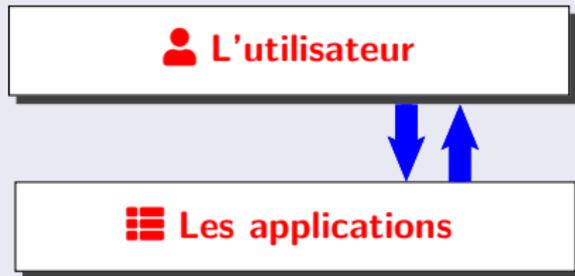
 **L'utilisateur**

La place du système d'exploitation

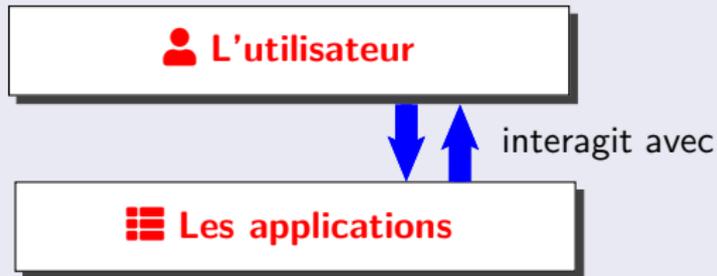
 **L'utilisateur**

 **Les applications**

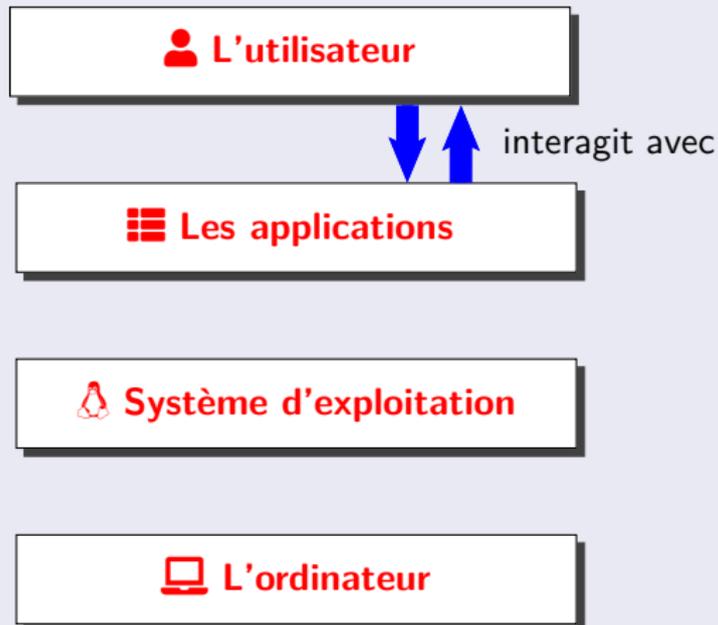
La place du système d'exploitation



La place du système d'exploitation

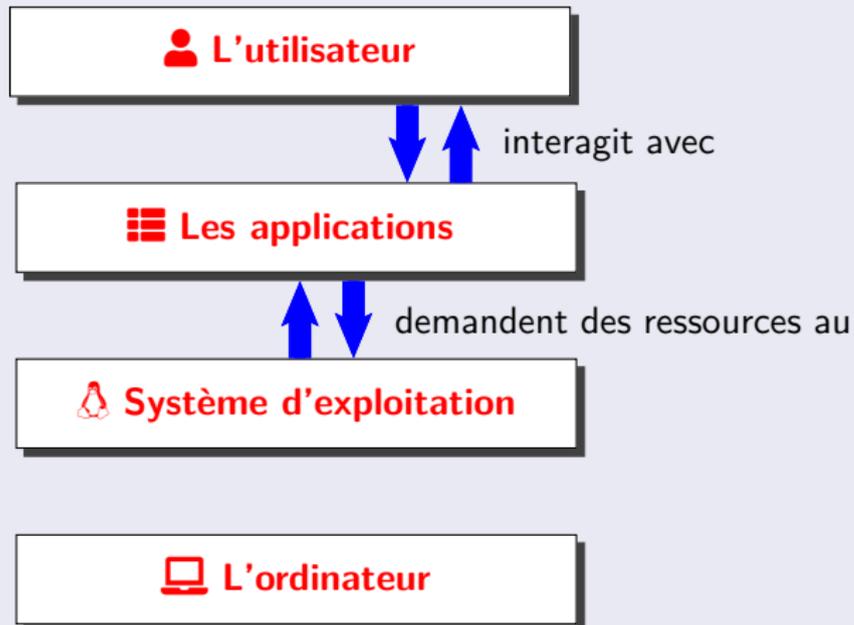


La place du système d'exploitation



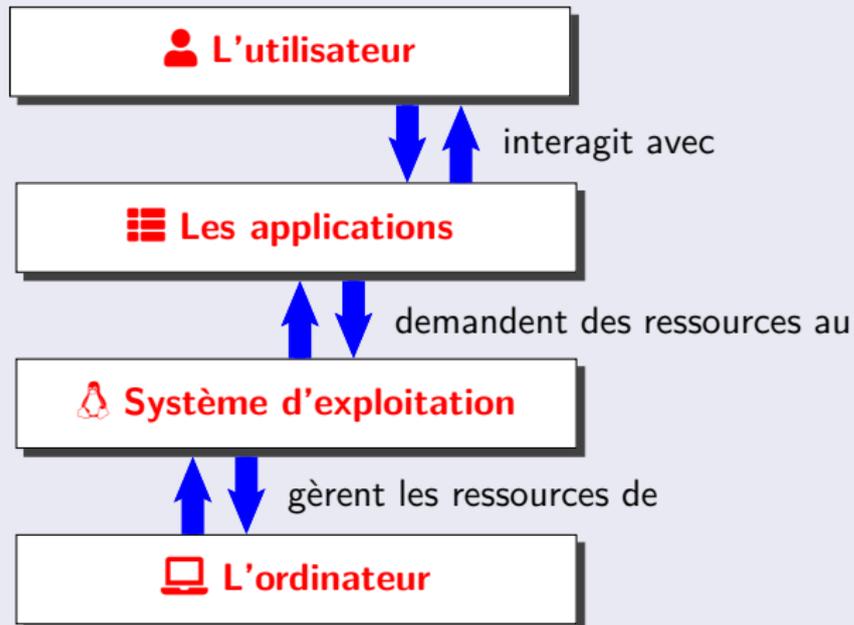
C1 Système d'exploitation

La place du système d'exploitation



C1 Système d'exploitation

La place du système d'exploitation



Fonctionnalités d'un système d'exploitation

Parmi les principales fonctionnalités d'un système d'exploitation, on peut citer :

Fonctionnalités d'un système d'exploitation

Parmi les principales fonctionnalités d'un système d'exploitation, on peut citer :

- La gestion des périphériques

Fonctionnalités d'un système d'exploitation

Parmi les principales fonctionnalités d'un système d'exploitation, on peut citer :

- La gestion des périphériques
- La gestion des fichiers

Fonctionnalités d'un système d'exploitation

Parmi les principales fonctionnalités d'un système d'exploitation, on peut citer :

- La gestion des périphériques
- La gestion des fichiers
- La gestion des ressources comme par exemple la mémoire ou l'unité de calcul (CPU)

Fonctionnalités d'un système d'exploitation

Parmi les principales fonctionnalités d'un système d'exploitation, on peut citer :

- La gestion des périphériques
- La gestion des fichiers
- La gestion des ressources comme par exemple la mémoire ou l'unité de calcul (CPU)
- La gestion (et récupération) des erreurs

Fonctionnalités d'un système d'exploitation

Parmi les principales fonctionnalités d'un système d'exploitation, on peut citer :

- La gestion des périphériques
- La gestion des fichiers
- La gestion des ressources comme par exemple la mémoire ou l'unité de calcul (CPU)
- La gestion (et récupération) des erreurs
- La sécurité des données

Du Code source à l'exécutable

Code source

- 1 Le **code source** est écrit par des développeurs informatique. Ce « code » est lisible et compréhensible par un être humain.

Du Code source à l'exécutable



- 1 Le **code source** est écrit par des développeurs informatique. Ce « code » est lisible et compréhensible par un être humain.
- 2 Ce code est **compilé**, c'est à dire qu'il est traduit par un programme informatique appelé compilateur.

Du Code source à l'exécutable



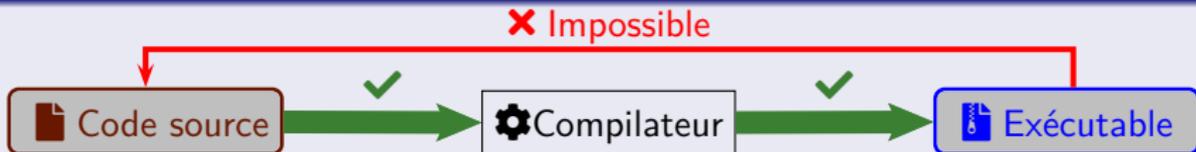
- 1 Le **code source** est écrit par des développeurs informatique. Ce « code » est lisible et compréhensible par un être humain.
- 2 Ce code est **compilé**, c'est à dire qu'il est traduit par un programme informatique appelé compilateur.
- 3 Le résultat obtenu est **un exécutable** fichier binaire compréhensible uniquement par un ordinateur

Du Code source à l'exécutable



- 1 Le **code source** est écrit par des développeurs informatique. Ce « code » est lisible et compréhensible par un être humain.
- 2 Ce code est **compilé**, c'est à dire qu'il est traduit par un programme informatique appelé compilateur.
- 3 Le résultat obtenu est **un exécutable** fichier binaire compréhensible uniquement par un ordinateur
- 4 L'opération inverse (passé de l'exécutable au code source) est virtuellement **impossible** !

Du Code source à l'exécutable



- 1 Le **code source** est écrit par des développeurs informatique. Ce « code » est lisible et compréhensible par un être humain.
- 2 Ce code est **compilé**, c'est à dire qu'il est traduit par un programme informatique appelé compilateur.
- 3 Le résultat obtenu est **un exécutable** fichier binaire compréhensible uniquement par un ordinateur
- 4 L'opération inverse (passé de l'exécutable au code source) est virtuellement **impossible** !

Libres et propriétaires

On distingue généralement :

- Les logiciels (et donc les OS) **propriétaires** développés par une société dans un but commercial. Par exemple Windows ou MacOS.

Libres et propriétaires

On distingue généralement :

- Les logiciels (et donc les OS) **propriétaires** développés par une société dans un but commercial. Par exemple Windows ou MacOS. Les utilisateurs n'ont alors pas le droit ni de modifier, ni de revendre le système d'exploitation.

Libres et propriétaires

On distingue généralement :

- Les logiciels (et donc les OS) **propriétaires** développés par une société dans un but commercial. Par exemple Windows ou MacOS. Les utilisateurs n'ont alors pas le droit ni de modifier, ni de revendre le système d'exploitation. En particulier, ils n'ont pas accès au **code source**. Seul l'exécutable est fourni.

Libres et propriétaires

On distingue généralement :

- Les logiciels (et donc les OS) **propriétaires** développés par une société dans un but commercial. Par exemple Windows ou MacOS. Les utilisateurs n'ont alors pas le droit ni de modifier, ni de revendre le système d'exploitation. En particulier, ils n'ont pas accès au **code source**. Seul l'exécutable est fourni.
- Les logiciels (et donc les OS) **libres** développés par une communauté d'informaticiens. Par exemple Linux.

Libres et propriétaires

On distingue généralement :

- Les logiciels (et donc les OS) **propriétaires** développés par une société dans un but commercial. Par exemple Windows ou MacOS. Les utilisateurs n'ont alors pas le droit ni de modifier, ni de revendre le système d'exploitation. En particulier, ils n'ont pas accès au **code source**. Seul l'exécutable est fourni.
- Les logiciels (et donc les OS) **libres** développés par une communauté d'informaticiens. Par exemple Linux. Le code source est alors fourni (*open source*), parfois avec des droits de modification.

Quelques repères historiques

- **1970–1990**

Développement du système UNIX (laboratoire Bells) par notamment Ken Thomson et Dennis Ritchie.

Quelques repères historiques

- **1970–1990**

Développement du système UNIX (laboratoire Bells) par notamment Ken Thomson et Dennis Ritchie.

- **1980–1990**

Développement du système MS-DOS (Microsoft)

Quelques repères historiques

- **1970–1990**

Développement du système UNIX (laboratoire Bells) par notamment Ken Thomson et Dennis Ritchie.

- **1980–1990**

Développement du système MS-DOS (Microsoft)

- **1983**

Projet de création d'un système d'exploitation libre semblable à UNIX (Richard Stallman). C'est le projet GNU.

Quelques repères historiques

- **1970–1990**

Développement du système UNIX (laboratoire Bells) par notamment Ken Thomson et Dennis Ritchie.

- **1980–1990**

Développement du système MS-DOS (Microsoft)

- **1983**

Projet de création d'un système d'exploitation libre semblable à UNIX (Richard Stallman). C'est le projet GNU.

- **1990–**

Développement progressif de Windows (Microfost)

Quelques repères historiques

- **1991**

Linus Torvalds alors étudiant se lance dans le développement d'un système d'exploitation *open source*.

Quelques repères historiques

- **1991**

Linus Torvalds alors étudiant se lance dans le développement d'un système d'exploitation *open source*.

Quelques repères historiques

- **1991**
Linus Torvalds alors étudiant se lance dans le développement d'un système d'exploitation *open source*.
- **1992–**
Développement rapide de Linux qui est associé à des applications du projet GNU. On devrait donc parler de GNU-Linux.

Quelques repères historiques

- **1991**
Linus Torvalds alors étudiant se lance dans le développement d'un système d'exploitation *open source*.
- **1992–**
Développement rapide de Linux qui est associé à des applications du projet GNU. On devrait donc parler de GNU-Linux.
- **2001–**
Apple démarre le développement de MacOS sur la base du système BSD, lui-même une variante d'UNIX.

Quelques repères historiques

- **1991**
Linus Torvalds alors étudiant se lance dans le développement d'un système d'exploitation *open source*.
- **1992–**
Développement rapide de Linux qui est associé à des applications du projet GNU. On devrait donc parler de GNU-Linux.
- **2001–**
Apple démarre le développement de MacOS sur la base du système BSD, lui-même une variante d'UNIX.
- **2008–**
Google crée et diffuse le système d'exploitation Android pour téléphone. Ce système utilise le noyau du système Linux.

Un état des lieux en 2020

- Windows est présent sur une très grande majorité des ordinateurs personnels ($\simeq 85\%$), suivi de MacOS ($\simeq 13\%$). Linux étant extrêmement minoritaire ($\simeq 2\%$).

Un état des lieux en 2020

- Windows est présent sur une très grande majorité des ordinateurs personnels ($\simeq 85\%$), suivi de MacOS ($\simeq 13\%$). Linux étant extrêmement minoritaire ($\simeq 2\%$).
- Sur les téléphones portables, c'est Android (et donc Linux) qui domine largement (environ 80% de part de marché).

Un état des lieux en 2020

- Windows est présent sur une très grande majorité des ordinateurs personnels ($\simeq 85\%$), suivi de MacOS ($\simeq 13\%$). Linux étant extrêmement minoritaire ($\simeq 2\%$).
- Sur les téléphones portables, c'est Android (et donc Linux) qui domine largement (environ 80% de part de marché).
- Dans le domaine des téléviseurs ou objet connectés, des serveurs web c'est Linux une fois de plus qui domine.

Un état des lieux en 2020

- Windows est présent sur une très grande majorité des **ordinateurs personnels** ($\simeq 85\%$), suivi de MacOS ($\simeq 13\%$). Linux étant extrêmement minoritaire ($\simeq 2\%$).
- Sur les téléphones portables, c'est Android (et donc Linux) qui domine largement (environ 80% de part de marché).
- Dans le domaine des téléviseurs ou objet connectés, des serveurs web c'est Linux une fois de plus qui domine.
- Enfin, Linux fait fonctionner la **totalité** des 500 ordinateurs les plus puissants du monde (source : <https://www.top500.org/statistics/list/>)

Des ressources vidéo

- Une  vidéo sur l'histoire des systèmes d'exploitation et leurs rôles.

Des ressources vidéo

- Une  vidéo sur l'histoire des systèmes d'exploitation et leurs rôles.
- Une  vidéo sur les notions de compilateurs/interpréteur.

Des ressources vidéo

- Une  vidéo sur l'histoire des systèmes d'exploitation et leurs rôles.
- Une  vidéo sur les notions de compilateurs/interpréteur.
- Une  vidéo sur Linux et son développement.

Interface en ligne de commande (CLI)

- Avant l'avènement des interfaces graphiques (GUI en anglais pour *Graphical User Interface*) et de la souris que nous connaissons aujourd'hui, l'utilisateur communiquait avec les applications (et donc aussi l'OS) par l'intermédiaire d'un simple clavier et d'une **interface en ligne de commande** (CLI en anglais pour *Command Line Interface*).

Interface en ligne de commande (CLI)

- Avant l'avènement des interfaces graphiques (GUI en anglais pour *Graphical User Interface*) et de la souris que nous connaissons aujourd'hui, l'utilisateur communiquait avec les applications (et donc aussi l'OS) par l'intermédiaire d'un simple clavier et d'une **interface en ligne de commande** (CLI en anglais pour *Command Line Interface*).
- Aujourd'hui encore et pour diverses raisons (contrôle plus fin de l'ordinateur, récupération d'erreurs, ...) la ligne de commande reste très utilisée.

CLI : manipulation des dossiers

`pwd` permet d'afficher le chemin complet du dossier dans lequel on se trouve.

CLI : manipulation des dossiers

- `pwd` permet d'afficher le chemin complet du dossier dans lequel on se trouve.
- `cd` permet de changer le dossier courant, on indique le dossier de destination :

CLI : manipulation des dossiers

- `pwd` permet d'afficher le chemin complet du dossier dans lequel on se trouve.
- `cd` permet de changer le dossier courant, on indique le dossier de destination :
- de façon absolue, c'est à dire depuis la racine du système de fichier
 - de façon relative, c'est à dire depuis le dossier courant, dans ce cas « .. » indique le dossier parent.

CLI : manipulation des dossiers

`pwd` permet d'afficher le chemin complet du dossier dans lequel on se trouve.

`cd` permet de changer le dossier courant, on indique le dossier de destination :

- de façon absolue, c'est à dire depuis la racine du système de fichier
- de façon relative, c'est à dire depuis le dossier courant, dans ce cas « .. » indique le dossier parent.

`mkdir` permet de créer un dossier

CLI : manipulation des dossiers

`pwd` permet d'afficher le chemin complet du dossier dans lequel on se trouve.

`cd` permet de changer le dossier courant, on indique le dossier de destination :

- de façon absolue, c'est à dire depuis la racine du système de fichier
- de façon relative, c'est à dire depuis le dossier courant, dans ce cas « .. » indique le dossier parent.

`mkdir` permet de créer un dossier

`rmdir` permet d'effacer un dossier vide

CLI : manipulation des dossiers

`pwd` permet d'afficher le chemin complet du dossier dans lequel on se trouve.

`cd` permet de changer le dossier courant, on indique le dossier de destination :

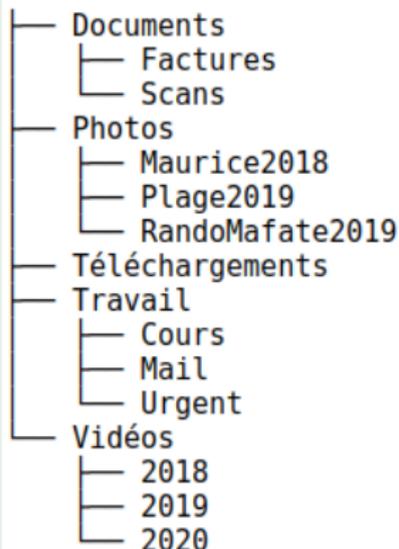
- de façon absolue, c'est à dire depuis la racine du système de fichier
- de façon relative, c'est à dire depuis le dossier courant, dans ce cas « .. » indique le dossier parent.

`mkdir` permet de créer un dossier

`rmdir` permet d'effacer un dossier vide

`mv` permet de renommer ou de déplacer un dossier (fonctionne aussi sur les fichiers)

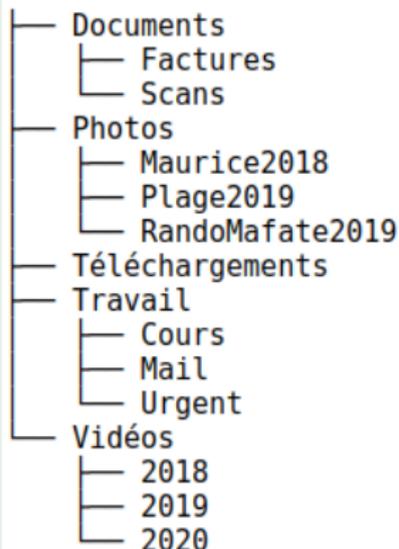
Exemples



On se trouve dans le dossier Cours, on utilise le chemin relatif, écrire les commandes pour

- 1 se déplacer vers le dossier Documents

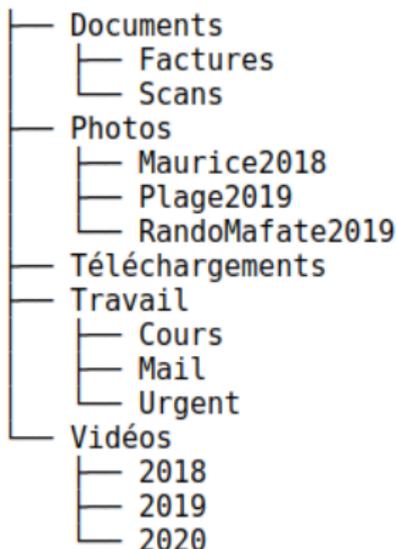
Exemples



On se trouve dans le dossier Cours, on utilise le chemin relatif, écrire les commandes pour

- 1 se déplacer vers le dossier Documents

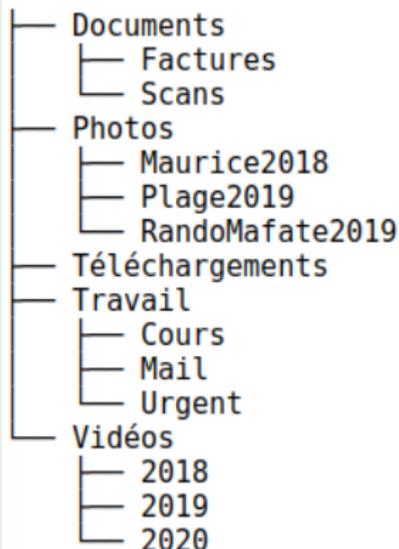
Exemples



On se trouve dans le dossier Cours, on utilise le chemin relatif, écrire les commandes pour

- 1 se déplacer vers le dossier Documents

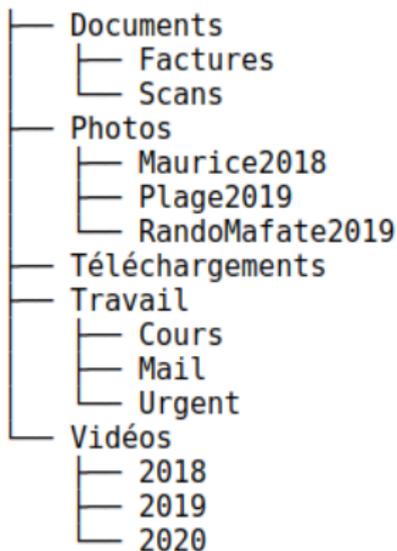
Exemples



On se trouve dans le dossier Cours, on utilise le chemin relatif, écrire les commandes pour

- 1 se déplacer vers le dossier Documents

Exemples



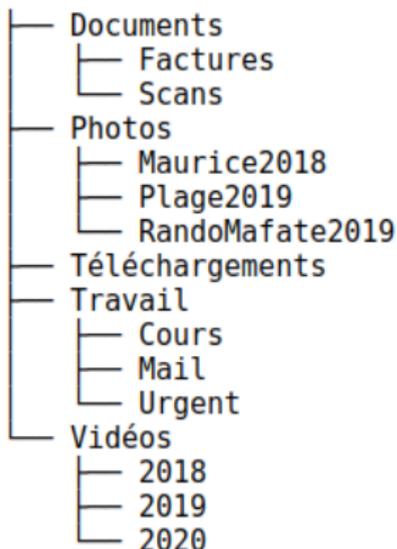
On se trouve dans le dossier Cours, on utilise le chemin relatif, écrire les commandes pour

- 1 se déplacer vers le dossier Documents

```
cd ../../Documents
```

- 2 y créer un dossier Important

Exemples



On se trouve dans le dossier Cours, on utilise le chemin relatif, écrire les commandes pour

- 1 se déplacer vers le dossier Documents

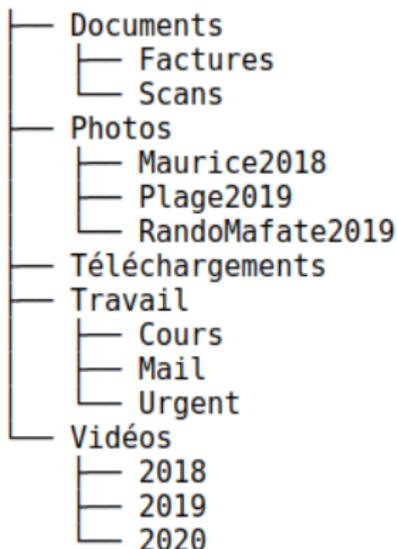
```
cd ../../Documents
```

- 2 y créer un dossier Important

```
mkdir Important
```

- 3 se déplacer vers le dossier Vidéos

Exemples



On se trouve dans le dossier Cours, on utilise le chemin relatif, écrire les commandes pour

- 1 se déplacer vers le dossier Documents

```
cd ../../Documents
```

- 2 y créer un dossier Important

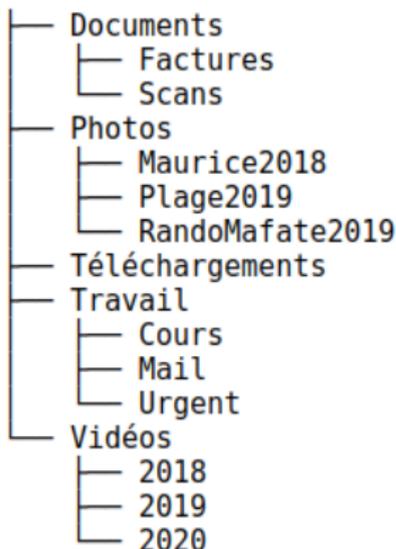
```
mkdir Important
```

- 3 se déplacer vers le dossier Vidéos

```
cd ../Vidéos
```

- 4 Supprimer le dossier 2018 (supposé vide)

Exemples



On se trouve dans le dossier Cours, on utilise le chemin relatif, écrire les commandes pour

- 1 se déplacer vers le dossier Documents

```
cd ../../Documents
```

- 2 y créer un dossier Important

```
mkdir Important
```

- 3 se déplacer vers le dossier Vidéos

```
cd ../Vidéos
```

- 4 Supprimer le dossier 2018 (supposé vide)

```
rmdir 2018
```

CLI : manipulation des fichiers

CLI : manipulation des fichiers

`ls` permet de lister le contenu d'un dossier, parmi les options les plus courantes on trouve :

CLI : manipulation des fichiers

`ls` permet de lister le contenu d'un dossier, parmi les options les plus courantes on trouve :

`ls -l` pour voir les droits sur les fichiers

`ls -a` pour voir les fichiers cachés, c'est à dire ceux dont le nom commence par un point .

CLI : manipulation des fichiers

`ls` permet de lister le contenu d'un dossier, parmi les options les plus courantes on trouve :

`ls -l` pour voir les droits sur les fichiers

`ls -a` pour voir les fichiers cachés, c'est à dire ceux dont le nom commence par un point .

`cat` permet de visualiser le contenu d'un fichier texte

CLI : manipulation des fichiers

`ls` permet de lister le contenu d'un dossier, parmi les options les plus courantes on trouve :

`ls -l` pour voir les droits sur les fichiers

`ls -a` pour voir les fichiers cachés, c'est à dire ceux dont le nom commence par un point .

`cat` permet de visualiser le contenu d'un fichier texte

`touch` permet de créer un fichier vide

CLI : manipulation des fichiers

`ls` permet de lister le contenu d'un dossier, parmi les options les plus courantes on trouve :

`ls -l` pour voir les droits sur les fichiers

`ls -a` pour voir les fichiers cachés, c'est à dire ceux dont le nom commence par un point .

`cat` permet de visualiser le contenu d'un fichier texte

`touch` permet de créer un fichier vide

`rm` permet d'effacer un fichier

`cp` permet de copier un fichier

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers,

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

Trois type de droits sont définis :

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

Trois type de droits sont définis :

- r** droit de lecture du fichier

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

Trois type de droits sont définis :

- r** droit de lecture du fichier
- w** droit d'écriture dans le fichier

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

Trois type de droits sont définis :

- r droit de lecture du fichier
- w droit d'écriture dans le fichier
- x droit d'exécution du fichier

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

Trois type de droits sont définis :

- r** droit de lecture du fichier
- w** droit d'écriture dans le fichier
- x** droit d'exécution du fichier

Ces droits sont définis pour :

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

Trois type de droits sont définis :

- r** droit de lecture du fichier
- w** droit d'écriture dans le fichier
- x** droit d'exécution du fichier

Ces droits sont définis pour :

- u** le propriétaire du fichier

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

Trois type de droits sont définis :

- r** droit de lecture du fichier
- w** droit d'écriture dans le fichier
- x** droit d'exécution du fichier

Ces droits sont définis pour :

- u** le propriétaire du fichier
- g** le groupe du fichier

CLI : gestion des permissions

Les systèmes de type Linux, conçu depuis l'origine pour être **multi-utilisateurs** possèdent un système de gestion des **permissions** sur les fichiers, assurant la sécurité du système et la protection des données des utilisateurs.

Trois type de droits sont définis :

- r** droit de lecture du fichier
- w** droit d'écriture dans le fichier
- x** droit d'exécution du fichier

Ces droits sont définis pour :

- u** le propriétaire du fichier
- g** le groupe du fichier
- o** tous les autres utilisateurs

C1 Système d'exploitation

CLI : command `chmod`

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` :

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxr-xr-x` :

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rwxr-xr-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxwxr-xr-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxwxr-xr-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :
 - `chmod g+w monfichier` :

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxwxr-xr-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :
 - `chmod g+w monfichier` : Ajoute (+) au groupe (g) le droit d'écriture (w)

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxrx-rx-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :
 - `chmod g+w monfichier` : Ajoute (+) au groupe (g) le droit d'écriture (w)
 - `chmod u+x monfichier` :

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxwxr-xr-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :
 - `chmod g+w monfichier` : Ajoute (+) au groupe (g) le droit d'écriture (w)
 - `chmod u+x monfichier` : Ajoute (+) au propriétaire (u) le droit d'exécution (x)

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxwxr-xr-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :
 - `chmod g+w monfichier` : Ajoute (+) au groupe (g) le droit d'écriture (w)
 - `chmod u+x monfichier` : Ajoute (+) au propriétaire (u) le droit d'exécution (x)
 - `chmod og-r monfichier` :

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxwxr-xr-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :
 - `chmod g+w monfichier` : Ajoute (+) au groupe (g) le droit d'écriture (w)
 - `chmod u+x monfichier` : Ajoute (+) au propriétaire (u) le droit d'exécution (x)
 - `chmod og-r monfichier` : Enlève (-) au groupe et aux autres (og) le droit de lecture (r)

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxwx-rx-x` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :
 - `chmod g+w monfichier` : Ajoute (+) au groupe (g) le droit d'écriture (w)
 - `chmod u+x monfichier` : Ajoute (+) au propriétaire (u) le droit d'exécution (x)
 - `chmod og-r monfichier` : Enlève (-) au groupe et aux autres (og) le droit de lecture (r)
 - `chmod a-r monfichier` :

CLI : command `chmod`

- L'affichage des droits sur un fichier se fait en affichant un tiret - si le droit est absent ou la lettre (r, w, x) désignant le droit sinon. On liste dans l'ordre les droits du propriétaire, puis ceux groupe puis ceux des autres. Par exemple :
 - `rw-r---` : L'utilisateur a les droits d'écriture et de lecture, le groupe a le droit de lecture, les autres n'ont aucun droit
 - `rxwx-rx-rx` : L'utilisateur a les droits d'écriture, de lecture et d'exécution, le groupe et les autres ont le droit de lecture et d'exécution
- La commande `chmod` permet de modifier les droits sur un fichier dont on est propriétaire. En voici quelques exemples :
 - `chmod g+w monfichier` : Ajoute (+) au groupe (g) le droit d'écriture (w)
 - `chmod u+x monfichier` : Ajoute (+) au propriétaire (u) le droit d'exécution (x)
 - `chmod og-r monfichier` : Enlève (-) au groupe et aux autres (og) le droit de lecture (r)
 - `chmod a-r monfichier` : Enlève (-) à tout le monde (a) le droit de lecture (r)