

Chapitre 1

Systèmes et lignes de commandes.

Sommaire.

1	Qu'est-ce qu'un système d'exploitation ?	1
1.1	Interpréteur de commandes (shell).	1
1.2	Organisation des données.	1
1.3	I-noeuds.	2

Les propositions marquées de ★ sont au programme de colles.

1 Qu'est-ce qu'un système d'exploitation ?

Définition 1: Système d'exploitation.

Un système d'exploitation est une interface entre l'utilisateur et le matériel informatique, il s'occupe d'isoler le code du matériel (noyau):

- Les changements de matériels sont transparents à l'utilisateur.
- Le système préserve l'intégrité des données.
- Il s'occupe de l'exécution de tous les programmes de façon équitable.

1.1 Interpréteur de commandes (shell).

Définition 2: Le shell.

Le shell est un processus qui permet de saisir des commandes et de les exécuter par le biais du terminal. Il existe plusieurs shells dont le bash, zsh, ... celui utilisé sera le **bash**.

Une commande est validée par un retour à la ligne.

Définition 3: Forme générale d'une commande.

- **Commande** : nom de la commande à exécuter.
- **Options** : précisent le comportement de la commande.
- **Arguments** : données sur lesquelles la commande agit.

On peut obtenir de l'aide sur une commande `cmd` en utilisant la commande `man cmd`.

1.2 Organisation des données.

Définition 4

Le stockage réel en machine est différent de la vision qu'en a l'utilisateur.

L'abstraction utilisée pour les données est de les regrouper dans des entités logiques, appelées fichiers:

- Fichiers réguliers : ceux contenant des données utilisables.
- Répertoires : ceux qui structurent les données.

Définition 5

Pour désigner un fichier, on utilise un chemin, c'est-à-dire la suite des étiquettes des répertoires dans l'arborescence des fichiers, séparées par des `/`.

Le chemin absolu est le chemin complet depuis la racine `/`, ou du répertoire de login `~`, le chemin relatif est le chemin depuis le répertoire courant.

Définition 6: Se déplacer dans l'arborescence.

Il existe des commandes pour se déplacer et se repérer dans l'arborescence:

- `pwd` : affiche le chemin absolu du répertoire courant.
- `ls` : liste les fichiers et répertoires du répertoire courant.
- `cd` : change de répertoire.

On peut ajouter l'option `-r` à `ls` pour lister les fichiers récursivement.

Définition 7: Modifier l'arborescence.

Il existe des commandes pour modifier l'arborescence:

- `mkdir` : crée un répertoire.
- `rmdir` : supprime un répertoire vide.
- `rm` : supprime un fichier ou un répertoire.
- `cp` : copie un fichier ou un répertoire.
- `mv` : déplace un fichier ou un répertoire.

1.3 I-noeuds.

Définition 8: I-noeud.

Un système de type UNIX / Linux identifie un fichier par une structure appelée **I-noeud**.

Un I-noeud contient les informations suivantes :

- Le type de fichier.
- La date de dernière modification.
- Le propriétaire du fichier.
- Les permissions d'accès.
- L'adresse des emplacements où sont stockées les données du fichier.

Un I-noeud ne contient **pas**:

- Le nom du fichier.
- Les données du fichier.

Définition 9: Lien avec l'arborescence.

Le lien entre le tableau d'I-noeuds et l'arborescence se fait via les répertoires.

Un répertoire est un fichier qui contient une liste de couples (nom, numéro I-noeud).

Sur un système UNIX, un fichier peut avoir plusieurs noms, il suffit que son numéro d'I-noeud apparaisse dans plusieurs couples (nom, numéro I-noeud).

Ces noms sont appelées des **liens** physiques sur le fichier. On peut en créer avec la commande `ln`.

Il existe aussi des liens symboliques, qui sont d'autres fichiers faisant le lien avec un fichier donné. On peut en créer avec la commande `ln -s`.

Définition 10: Groupes d'utilisateurs: droits.

Chaque fichier possède un utilisateur propriétaire, il possède également un groupe propriétaire.

On définit les droits d'un fichier séparément pour le propriétaire (u), le groupe propriétaire (g) et les autres utilisateurs (o).

Il y a trois types de droits, la lecture (r), l'écriture (w) et l'exécution (x).

On les modifie via la commande `chmod` : `chmod utilisateurs operateur droit fichier` (sans espaces).

Exemple: `chmod u+x fichier` ajoute le droit d'exécution au propriétaire du fichier.

Seuls root et le propriétaire d'un fichier peuvent modifier ces droits.