

Première NSI  
Chapitre IX - Les systèmes d'exploitation

# I. Fonctions d'un système d'exploitation

En informatique, un **système d'exploitation** (souvent appelé OS — de l'anglais Operating System) est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs.

Il reçoit des demandes d'utilisation des ressources de l'ordinateur de la part de ces logiciels applicatifs et gère ces demandes ainsi que les ressources nécessaires, évitant les interférences entre les logiciels.

Le système d'exploitation est donc composé de programmes qui activent des **processus**. On peut exécuter ces programmes via le **shell** et son invite de commande.

De plus, le système d'exploitation gère le **système de fichiers** et donc les **droits** et les **permissions** des utilisateurs.

Il existe différents types de systèmes d'exploitation mais on peut les regrouper dans deux familles : les systèmes d'exploitation libres et les systèmes d'exploitation propriétaires.

## Le système d'exploitation propriétaire

La création d'un tel logiciel est une oeuvre collective donc le droit patrimonial est donné à l'éditeur (c'est-à-dire l'employeur de ceux qui ont créé le logiciel). L'éditeur peut donc utiliser ses droits comme bon lui semble et est protégé par le droit d'auteur (copyright).

L'utilisateur est lié à l'éditeur par un contrat appelé licence.

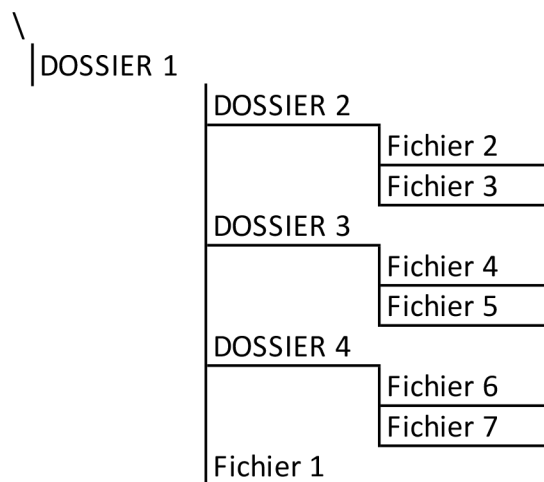
## Le système d'exploitation libre

Le logiciel libre permet de :

- l'utiliser sans restriction
- l'étudier et le modifier
- le redistribuer
- distribuer des version modifiées

# II. Système de fichiers

On peut représenter le système de fichiers par une arborescence.



La racine est noté « / ». Elle est son propre parent.

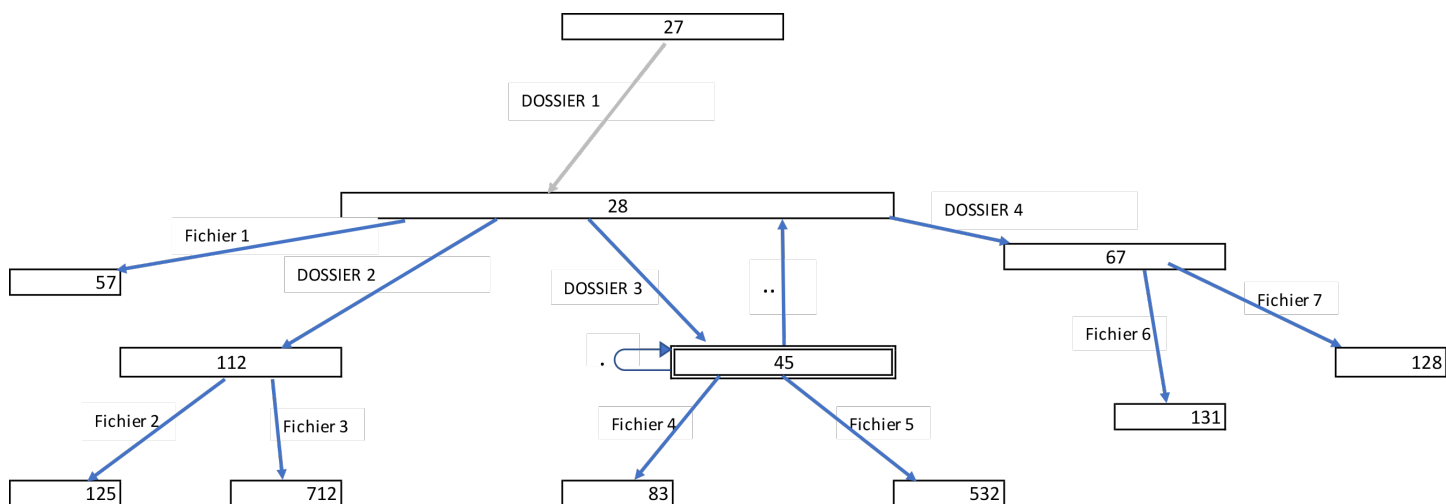
Les noeuds non terminaux sont des répertoires.

Les noeuds terminaux sont les fichiers standards et contiennent des données.

### Remarque

Les répertoires ne sont que des fichiers dans lesquels il y a la liste des fichiers qu'il contient.

Dans les faits, la représentation la plus juste d'un système de fichiers est le suivant :



Les répertoires ont toujours au moins deux fils : « . » (le nœud lui-même) et « .. » (son parent) (voir le nœud 45). Les arcs sont nommés par une chaîne de caractères (sauf \0 et /). Les numéros dans les nœuds sont des liens entre le nommage du fichier et le contenu. Tous ces nœuds sont appelés **Inoeds** (Inode).

On appelle **chemin absolu** d'un fichier l'ensemble des nœuds parcourus depuis la racine jusqu'au fichier. On appelle **chemin relatif** d'un fichier l'ensemble des nœuds parcourus depuis le dossier en cours jusqu'au fichier.

### III. Commandes de base

Toutes les commandes shell créent un processus qui possèdent une **entrée standard**, une **sortie standard** et une **sortie d'erreur**.

L'entrée standard est en général le clavier.

La sortie standard et la sortie d'erreur sont en général l'écran.

Liste non exhaustive des commandes linux

commande	commentaire
<code>&gt; fichier</code>	redirige la sortie standard sur le fichier
<code>&lt; fichier</code>	redirige l'entrée standard depuis le fichier
<code>&gt;&gt; fichier</code>	concatène la sortie standard au fichier
<code>2 &gt; fichier</code>	redirige la sortie d'erreur sur le fichier
<code>2 &gt;&gt; fichier</code>	concatène la sortie d'erreur au fichier
<code>cat</code>	édite du contenu
<code>cat fichier</code>	envoie dans la sortie standard le contenu du fichier
<code>cat fichier1 fichier2</code>	concatène les deux fichiers
<code>cd nomRepertoire</code>	se déplace dans un répertoire
<code>cp fichier repertoire</code>	copie le fichier dans le répertoire
<code>echo chaineDeCaractres</code>	envoie dans la sortie standard la chaîne de caractère
<code>ls</code>	liste le contenu du dossier courant
<code>mkdir nomDuRepertoire</code>	crée un répertoire
<code>mv fichier1 repertoire</code>	déplace le fichier1 dans le répertoire
<code>mv fichier1 fichier2</code>	renomme le fichier1 en fichier2
<code>mv repertoire1 repertoire2</code>	renomme le répertoire1 en repertoire2
<code>pwd</code>	affiche le chemin du répertoire courant depuis le répertoire personnel (s'il existe) ou depuis la racine
<code>rm fichier</code>	supprime le fichier
<code>rmdir repertoire</code>	supprime un répertoire vide
<code>touch fichier</code>	crée un fichier
<code>tree</code>	affiche l'arbre depuis le répertoire en cours

Pour plus d'information sur une commande, on peut se renseigner sur le site [Wiki Ubuntu](#).

#### Exemple

```
1 ~# ls
2 arborescence.pdf          arborescence.xlsx
3 arborescence.png         `systeme de fichiers.png
```

#### Remarque

- le symbole `~#` correspond à l'invite de commande.
- le symbole `..` représente le répertoire parent.
- le symbole `.` représente le répertoire courant.
- le symbole `*` représente une chaîne de caractère quelconque.

## IV. Droits et permissions

Lorsqu'un fichier (ou répertoire) est créé, des **droits d'accès** précisent qui peut le lire, le modifier ou l'exécuter. L'ensemble des droits définit les **permissions**. Cette notion de droit, est indispensable dans un système multi-utilisateurs afin de préserver la confidentialité de ses documents à chaque utilisateur.

### IV.1. Vérification des droits d'un fichier

On utilise la commande suivante :

```
1 ls -l nom_du_fichier
```

Voici les informations qu'on peut lire :

Type	Propriétaire	Groupe auquel appartient le propriétaire	Autres utilisateurs
1 caractère	3 caractères	3 caractères	3 caractères

#### Type

- d : répertoire (directory)
- - : fichier

#### Droits

- r : droit de lire le fichier ou d'examiner le répertoire
- w : droit d'écrire dans le fichier ou d'ajouter un répertoire dans un répertoire
- x : droit d'exécuter le fichier ou d'accéder au répertoire

### IV.2. Modifier les droits d'un fichier

On utilise la commande `chmod` :

Il y a deux façons d'utiliser la commande 'chmod' : la méthode symbolique et la méthode numérique.

#### 1. La méthode symbolique

```
1 chmod degre_propriete action acces nom_du_fichier
```

- degré de propriété
  - u : le propriétaire
  - g : utilisateurs appartenant au groupe du propriétaire
  - o : les autres
  - a : tout le monde
- action
  - + : ajoute un droit
  - - : enlève un droit
  - = définit un droit
- type d'accès
  - r : lecture
  - w : écriture
  - x : exécution

## Exemple

```
1 ~# ls -l nom_fichier
2 -rw-rw-rw- 1 rcaneri staff 13 16 août 18:53 nom_fichier
3 ~# chmod go-rw nom_fichier
4 ~# ls -l nom_fichier
5 -rw----- 1 rcaneri staff 13 16 août 18:53 nom_fichier
```

On remarque bien que la commande `chmod go-rw nom_fichier` enlève les droits en lecture et écriture sur le fichier `nom_fichier` pour les utilisateurs appartenant au groupe du propriétaire et aux autres.

## 2. La méthode numérique

La notation numérique permet via le passage à un codage binaire de représenter un bloc de permissions, sous forme d'un triplet de chiffres compris entre 0 et 7.

Chaque lettre utilisée dans la méthode symbolique (r, w, ou x) est symbolisée par un 1 et chaque symbole - est symbolisée par un 0. On convertit ensuite trois bits par trois bits binaires dans le système décimal, ce qui donne un nombre à 3 chiffres.

On utilise alors la commande suivante :

```
1 chmod 3chiffres nom_du_fichier
```

## Exemple

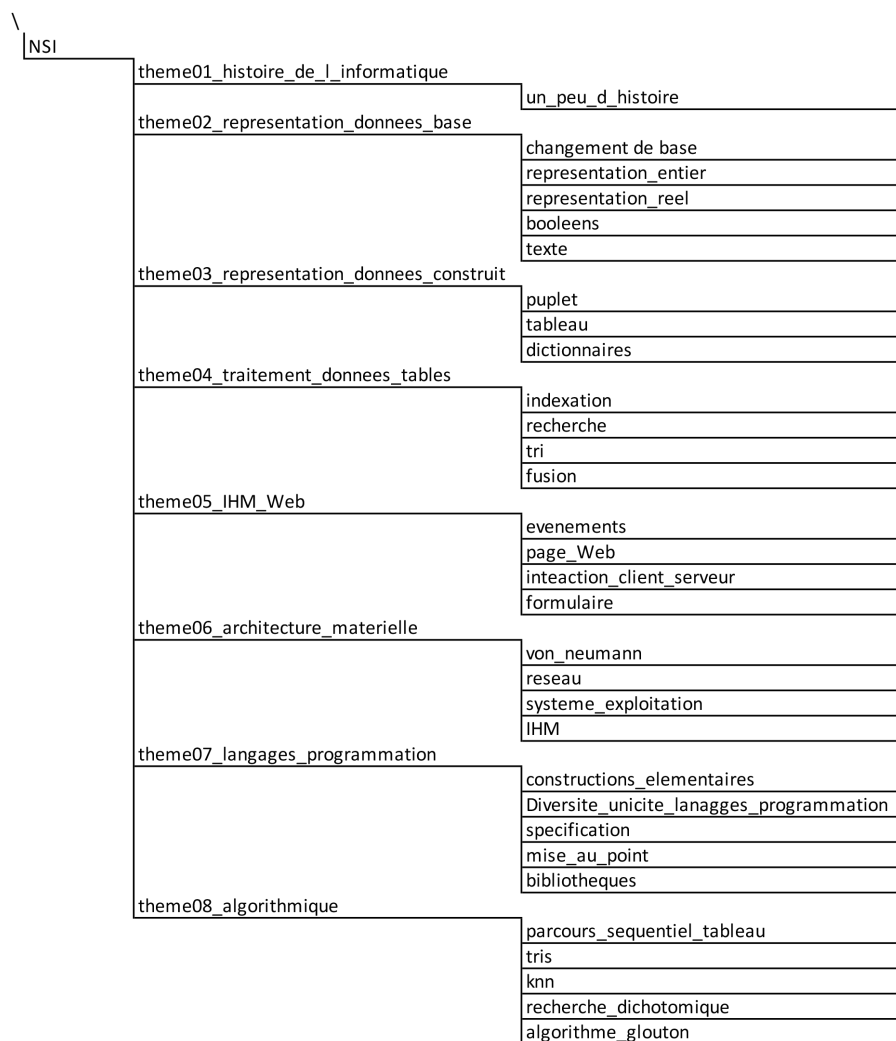
```
1 ~# ls -l nom_fichier
2 -rw-rw-rw- 1 rcaneri staff 13 16 août 18:53 nom_fichier
3 ~# chmod 600 nom_fichier
4 ~# ls -l nom_fichier
5 -rw----- 1 rcaneri staff 13 16 août 18:53 nom_fichier
```

On remarque que la commande `chmod 600 nom_fichier` enlève les droits en lecture et écriture sur le fichier `nom_fichier` pour les utilisateurs appartenant au groupe du propriétaire et aux autres.

## V. Exercices

### V.1. Exercice 1

1. Créer un dossier intitulé « NSI ».
2. Écrire les commandes d'instructions permettant d'obtenir l'arborescence ci-dessous :



3. Écrire dans le fichier `booléen` le texte suivant :  
Un booléen est une variable à deux états : vrai ou faux.
4. Changer les droits du fichier `tris` afin que les utilisateurs appartenant au groupe du propriétaire aient les droits en écriture et en lecture. *utiliser la méthode symbolique.*
5. Changer les droits du fichier `specification` afin que tout le monde ait les droits en lecture et en exécution. Mais seul le propriétaire aura les droit en écriture. *utiliser la méthode numérique.*