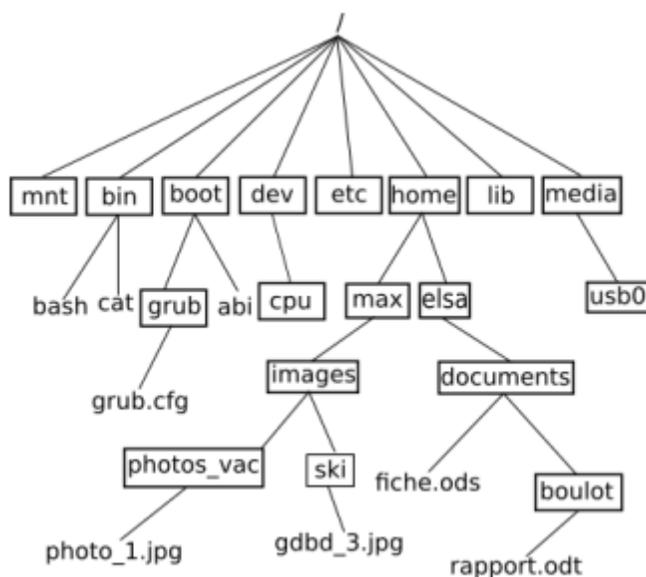


# arbresElevés

December 1, 2024

## 1 Structure arborescente - Recherche de fichier

%%html



*Structure hiérarchique arborescente d'un système Linux*

Pour retrouver un document, les systèmes d'exploitation proposent une fonction de recherche. Par exemple sous linux : `>find -name "mon-fichier.pdf"` >Ce qui signifie rechercherun fichier dont le nom est mon-fichier.pdf dans le répertoire courant et ses sous-répertoires

Objectif : Découvrir une nouvelle structure de données : les arbres.

### 1.1 Sommaire

1. Structure hiérarchique arborescente
2. Parcourir un arbre

### 1.2 Structure hiérarchique arborescente

#### 1.2.1 Définition

**À retenir** >Un arbre est défini par : >- un nœud particulier qui constitue la racine, >- plusieurs sous-ensembles d'autres arborescences reliées à la racine. \*\*\*

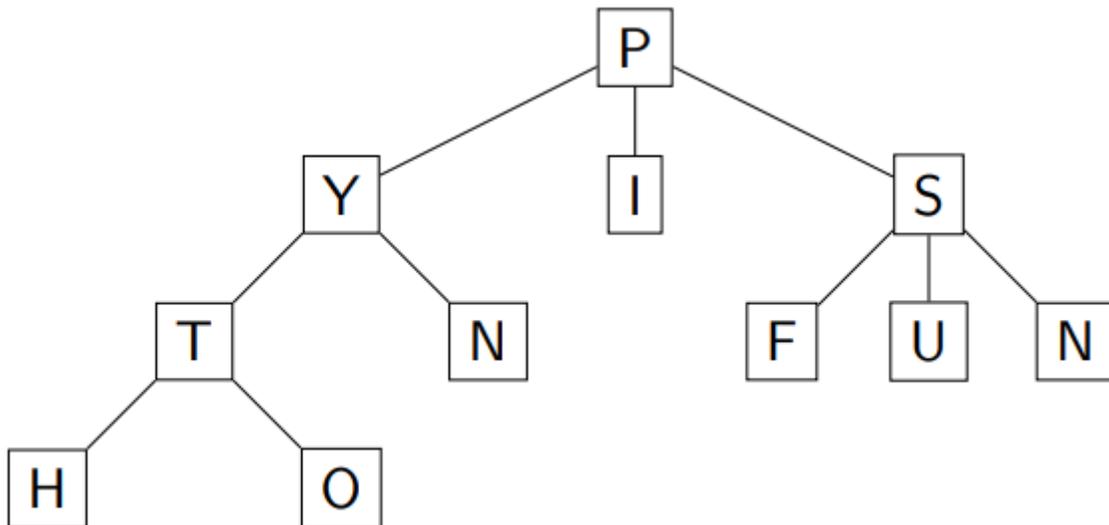
---

**Vocabulaire** >On nomme : >- **nœud-fils** l'ensemble des nœuds reliés à un même **nœud-père**, >- **feuilles** les nœuds qui n'ont pas de fils. \*\*\*

---

**Remarque**

>De manière usuelle un arbre est représentée à l'envers, la racine en haut. \*\*\*



*Une structure arborescente*

### 1.2.2 Caractéristiques

---

**À retenir** >La **taille** d'un arbre est le nombre de nœuds de la structure. \*\*\*

*La taille de l'arbre est 11.*

---

**À retenir** >La **hauteur (ou profondeur)** d'un arbre est la longueur du plus grand chemin entre la racine et une feuille. \*\*\*

*La hauteur de l'arbre est 3*

---

**Remarque**

>La définition de la hauteur varie dans la littérature. Elle peut être présentée comme le nombre maximum de sommets entre la racine et une feuille. La hauteur de l'arbre serait alors 4. \*\*\*

### 1.3 Parcourir un arbre

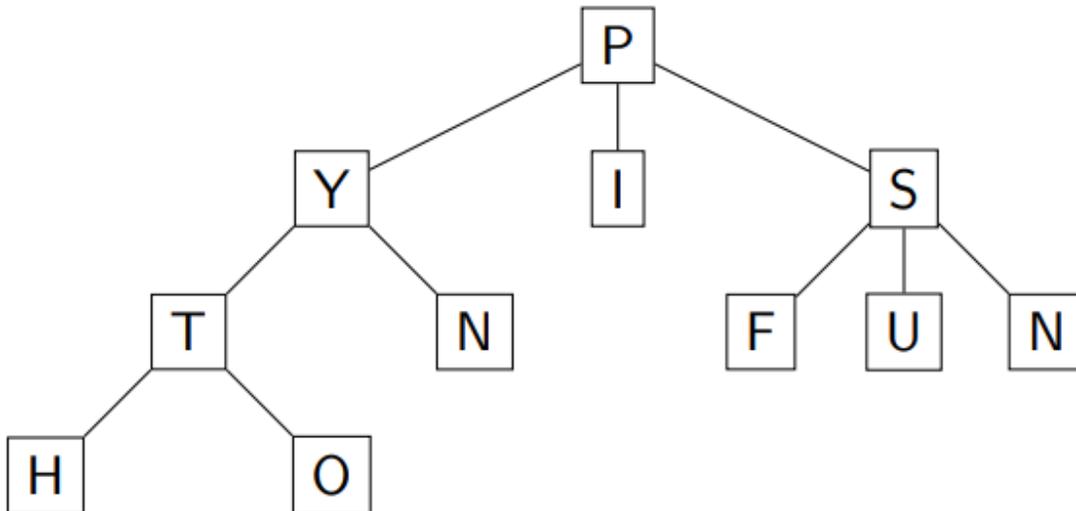
#### 1.3.1 Parcours en largeur

---

**À retenir** >L'arbre est parcouru niveau par niveau. À chaque étage les nœuds sont parcourus avant de passer au niveau suivant. L'ordre des nœuds par niveau n'est pas déterminé \*\*\*

#### 1.3.2 Activité 1

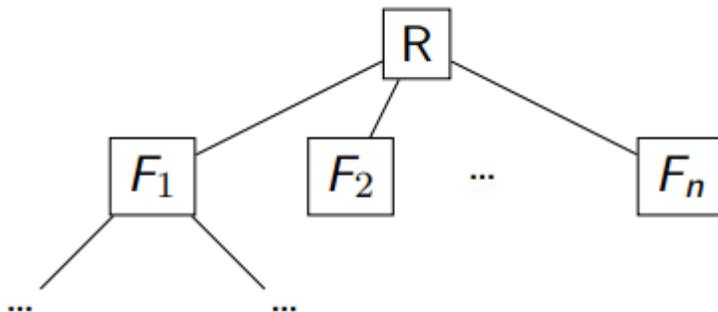
**Consigne** Parcourir en largeur l'arbre suivant.



#### 1.3.3 Parcours en profondeur

---

**À retenir** >Dans un parcours en profondeur, un des sous-arbres est parcouru entièrement avant qu'un autre ne soit exploré. C'est un algorithme récursif. \*\*\*



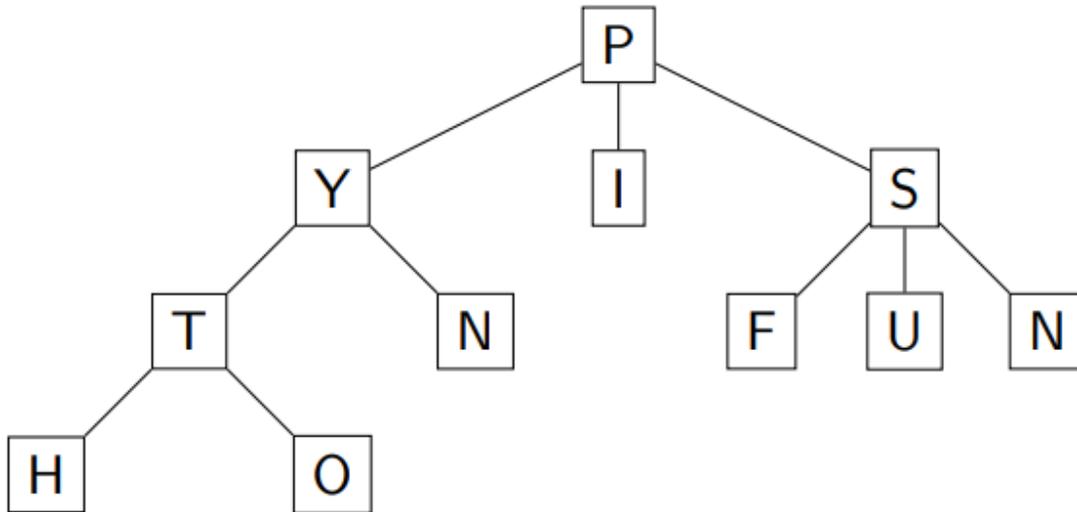
\* **À retenir** >On distingue trois parcours en profondeur : >- ordre préfixe : **On liste R puis les nœuds de F1 en ordre préfixe, puis les nœuds de F2 en ordre préfixe...** >- ordre infixe : **On liste les nœuds de F1 en ordre infixe, puis R, puis les nœuds de F2 en ordre infixe...** >- ordre

suffixe\*\* : On liste les nœuds de F1 en ordre suffixe, puis les nœuds de F2 en ordre suffixe..., puis R.

---

### 1.3.4 Activité 2

**Consigne** Parcourir en profondeur l'arbre suivant.



### 1.3.5 Rechercher un fichier

### 1.3.6 Activité 3

**Consigne**

1. Se rendre sur l'émulateur de [console Linux](#)
2. Créer l'arborescence de dossiers représentée par l'arbre, à l'aide des instructions suivantes :
  - > mkdir p # Créer le dossier p
  - > cd p # Entrer dans le dossier p
  - > cd .. # Retourner dans le dossier père

### 1.3.7 Activité 4

**Consigne**

1. Se placer dans le dossier P.
2. La commande suivante affiche le parcours d'une recherche quelconque. L'exécuter > find -print
3. Quel type de parcours effectue la fonction find ?