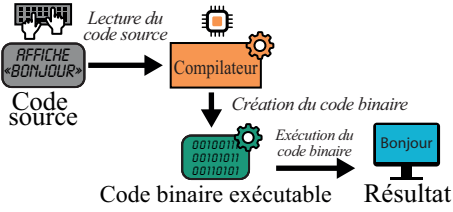
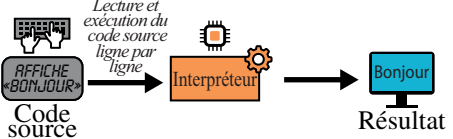
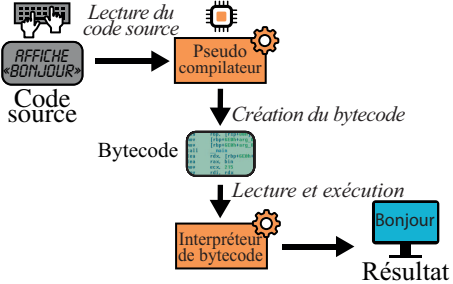


Types de programmation

● Exécution d'un programme

Il existe différents moyens d'exécuter un programme suivant le langage utilisé.

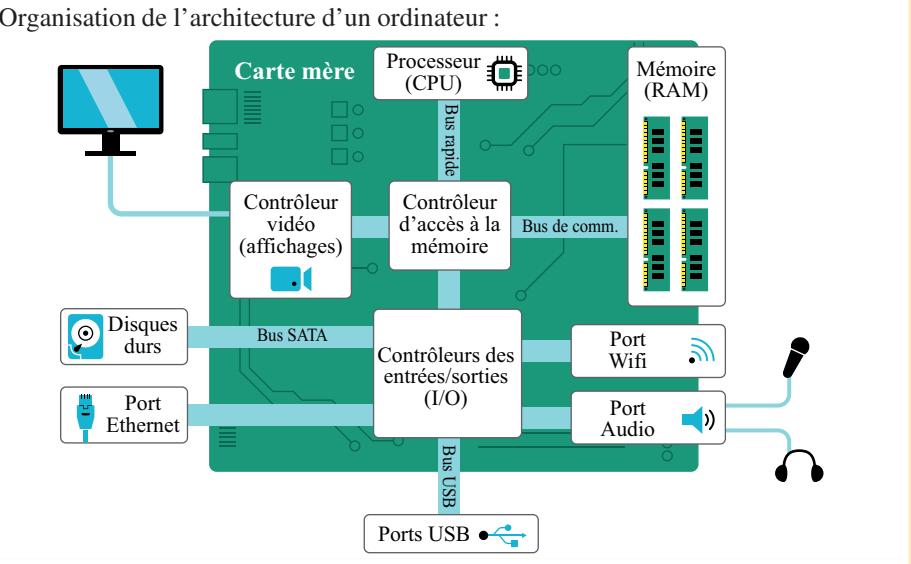
Type d'exécution	Représentation	Exemple
Compilation		C, C++, Pascal
Interprétation		Python, javascript
Semi-interprétation		Java

● Paradigmes de programmation

Un *paradigme* de programmation est un type de programmation.

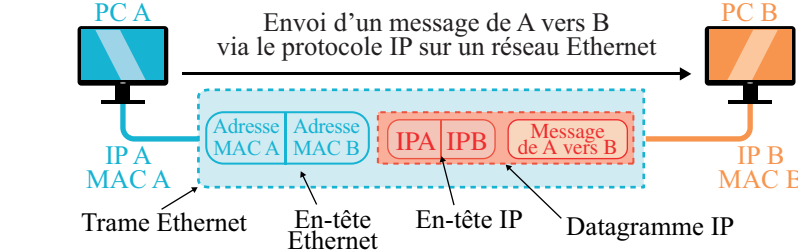
Type de programmation	Concepts utilisés	Avantages	Inconvénients	Exemple
Impérative	● Séquences d'instructions ● Variables	● Facile à apprendre ● Simplicité d'utilisation	● Effets de bord ● Peu structuré	Langage machine Langage C Fortran
Fonctionnelle	● Fonctions, éventuellement récursives ● Les fonctions sont aussi des données	● Structuré ● Pas d'effet de bord	● Difficile d'accès ● Complexe à utiliser	LISP Ocaml
Orientée objet	● les types (appelés <i>classes</i>) définissent une structure et un comportement	● Structuration forte ● Modularité ● Évolutivité	● Concepts sophistiqués ● Consommation de ressource importante	Java C++ C# Visual Basic

Architecture matérielle



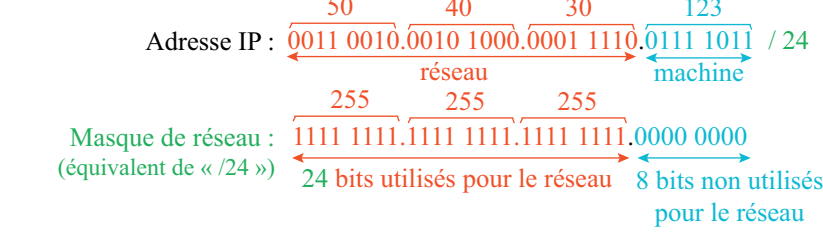
Réseaux

● Encapsulation des données via IP sur un réseau ethernet



● Masque de réseau

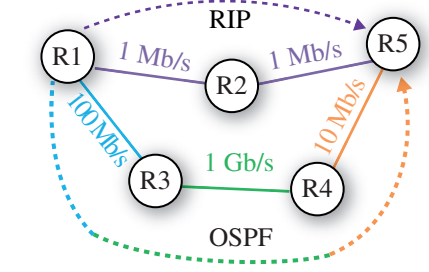
Décomposition d'une adresse IP en adresse de réseau et de machine.



● Protocoles de routage RIP et OSPF

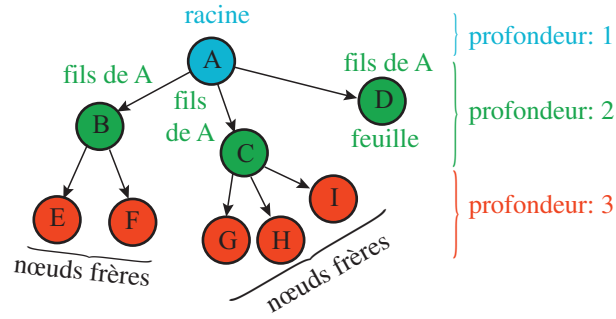
RIP (Routing Information Protocol)
Plus court chemin déterminé par le nombre de sauts
R1 → R2 → R5

OSPF (Open Short Path First)
Plus court chemin déterminé par le coût des routes
R1 → R3 → R4 → R5



Arbres

Arbre : structure de données prenant la forme d'un graphe orienté connexe dont tous les nœuds ont un unique nœud parent, à l'exception du nœud racine (qui n'en a aucun).

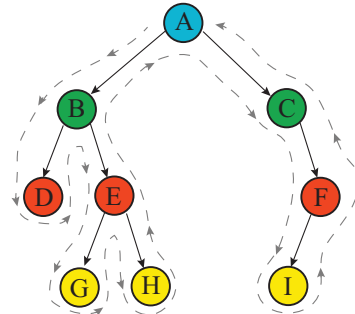


• Propriétés des arbres binaires

Soit un arbre de hauteur h à f feuilles et n nœuds.

$n \leq 2^{h+1} - 1$	$\log_2(n+1) - 1 \leq h \leq n$
$h \geq \log_2(f)$	$\log_2(f) \leq \log_2\left(\frac{n+1}{2}\right)$

• Parcours en profondeur



Ordre	Description	Liste des nœuds
Préfixe	Ordre dans lequel on rencontre les nœuds pour la première fois.	A - B - D - E - G - H - C - F - I
Postfixe	Ordre dans lequel on rencontre les nœuds pour la dernière fois.	D - G - H - E - B - I - F - C - A
Infixe	Chaque nœud ayant un fils gauche est répertorié la seconde fois qu'on le rencontre, et chaque nœud n'ayant pas de fils gauche est répertorié la première fois qu'on le rencontre.	D - B - G - E - H - A - C - I - F

• Parcours en largeur

On liste les nœuds suivant leur niveau dans l'arbre :



Graphes

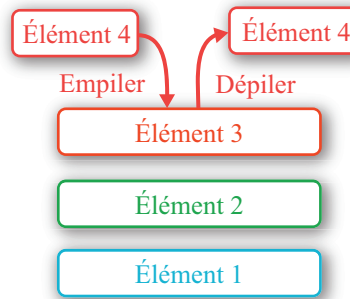
Graphe : structure de données relationnelle qui formalise les connexions entre plusieurs états.

	Graphes non pondérés	Graphes pondérés
Représentation sagittale		
Matrice d'adjacence ou de valuation	$M = \begin{pmatrix} A & B & C \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{matrix} A \\ B \\ C \end{matrix}$	$M = \begin{pmatrix} A & B & C \\ 0 & 5 & 7 \\ 5 & 0 & 2 \\ 7 & 2 & 0 \end{pmatrix} \begin{matrix} A \\ B \\ C \end{matrix}$
Implémentation de la matrice	$M = \begin{bmatrix} [0, 1, 1], \\ [1, 0, 1], \\ [1, 1, 0] \end{bmatrix}$	$M = \begin{bmatrix} [0, 5, 7], \\ [5, 0, 2], \\ [7, 2, 0] \end{bmatrix}$
Implémentation par dictionnaire (liste d'adjacence)	$G = \{ \begin{matrix} 'A' : ['B', 'C'], \\ 'B' : ['A', 'C'], \\ 'C' : ['A', 'B'] \end{matrix} \}$	$G = \{ \begin{matrix} 'A' : \{ 'B' : 5, 'C' : 7 \}, \\ 'B' : \{ 'A' : 5, 'C' : 2 \}, \\ 'C' : \{ 'A' : 7, 'B' : 2 \} \end{matrix} \}$

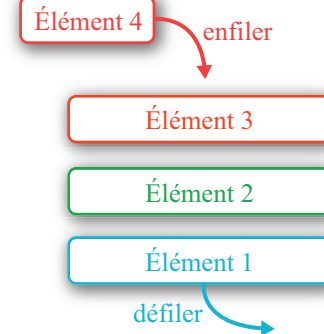
Structures de données élémentaires

Tableau	Séquence finie et de taille fixe d'éléments auxquels on peut accéder par leur indice.
Dictionnaire	Association dynamique de paires clé/valeur.
Liste	Groupe d'éléments ordonnés que l'on peut manipuler individuellement.
Pile	Collection remplie sur le principe « dernier arrivé, premier sorti » (LIFO).
File	Collection remplie sur le principe « premier arrivé, premier sorti » (FIFO).

• Pile

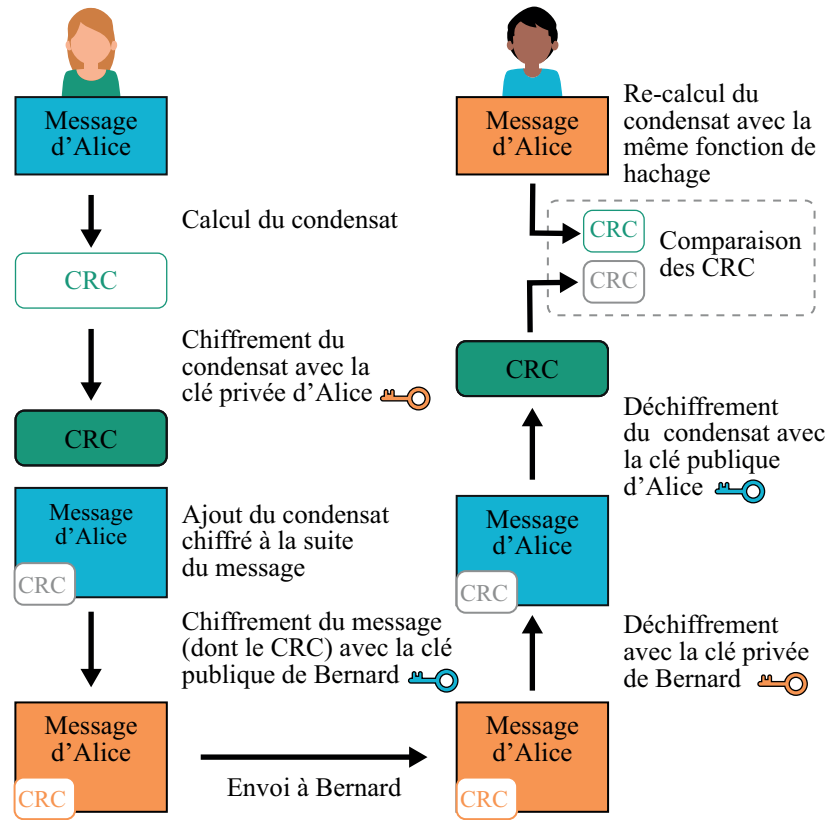


• File



Sécurité

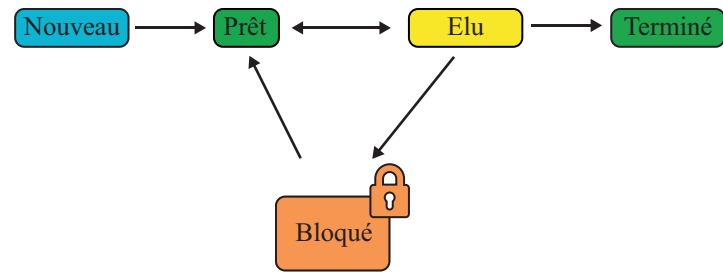
Principe de l'authentification d'un message entre deux correspondants :



Processus

Processus (*process* en anglais) : instance d'un programme, chargée en mémoire, qui est exécutée de manière continue ou discontinue par le système d'exploitation.

• États principaux d'un processus

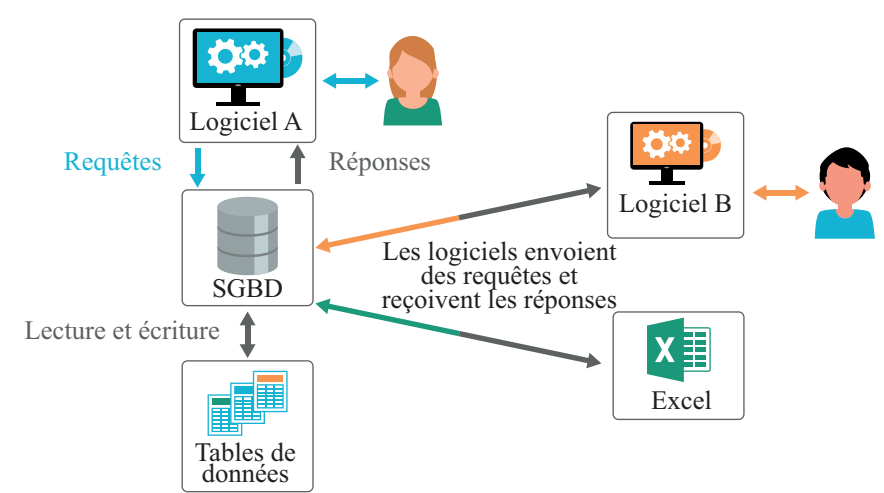


• Deadlock (interblocage)

Deux processus (A) et (B) sont chacun en attente d'un travail que doit fournir l'autre.

Système de gestion de bases de données (SGBD)

• Principe de fonctionnement d'un SGBD



• Les différentes jointures de tables

TABLE_A	Opérations de jointure	TABLE_B
* A_ID		* B_ID
AChamp1		BChamp1
AChamp2		BChamp2
AChamp3		BChamp3

Intersection	<code>select * from TABLE_A join TABLE_B as A as B on B.BChamp1=A.AChamp1</code>	
--------------	--	--

• Requêtes SQL

→ Requête de sélection :

`SELECT ... FROM <table> WHERE ... AND ...`

→ Requête d'insertion :

`INSERT INTO <table> VALUES ...`

→ Requête de suppression :

`DELETE FROM <table> WHERE ...`

→ Requête de modification :

`UPDATE <table> SET ... = ... WHERE ...`