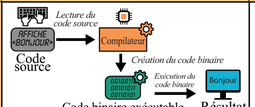
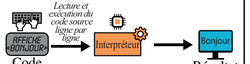
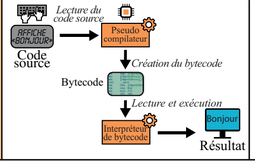
Types de programmation

• 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

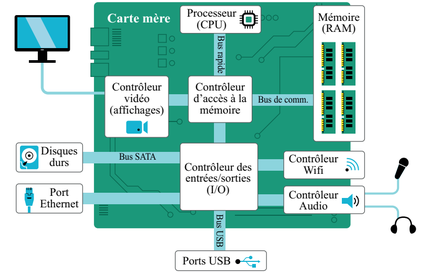
• Paradigmes de programmation

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

Type de programmation	Concepts utilisés	Avantages	Inconvénients	Exemple
Impérative	• Séquences d'instructions • Variables globales	• Facile à apprendre • Simplicité d'utilisation	• Effets de bord • Peu structuré	Langage machine Langage C Fortran
Fonctionnelle	• Fonctions, éventuellement récursives • Variables globales interdites	• Structuré • Pas d'effet de bord	• Difficile d'accès • Complexe à utiliser	LISP
Orientée objet	• Objets en interaction mutuelle, structurés en classes	• Modélisation proche du réel • Modularité • Évolutivité	• Concepts sophistiqués • Consommation de ressource importante	Java C++ C# Visual Basic

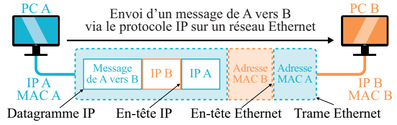
Architecture matérielle

Organisation de l'architecture d'un ordinateur :



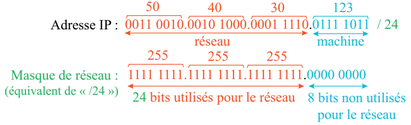
Réseaux

• Encapsulation des données via IP sur un réseau ethernet



• Masque de réseau

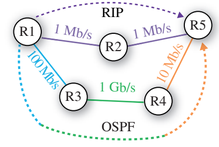
Décomposition d'une adresse IP en adresse de réseau et de machine.



• Protocoles de routage RIP et OSPF

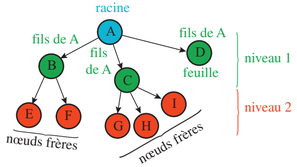
RIP (Routing Information Protocol)
Plus court chemin déterminé par le nombre de sauts
R1 → R2 → R5

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



Arbres

Arbre : structure de données représentant un graphe acyclique orienté connexe.

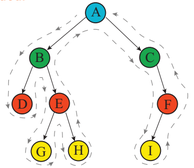


Propriétés

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

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

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

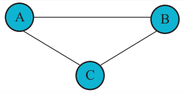
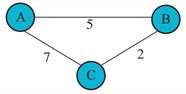
Parcours en largeur

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

$$\underbrace{A}_{\text{niveau 0}} - \underbrace{B - C}_{\text{niveau 1}} - \underbrace{D - E - F - G}_{\text{niveau 2}} - \underbrace{H - I}_{\text{niveau 3}}$$

Graphes

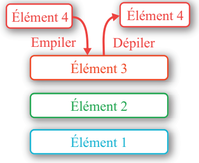
Graphe : structure de données relationnelle qui formalise les connexions entre plusieurs états.

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

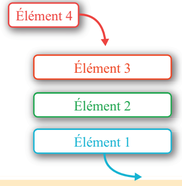
Structures de données élémentaires

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	Liste remplie sur le principe « dernier arrivé, premier sorti » (LIFO).
File	Liste remplie sur le principe « premier arrivé, premier sorti » (FIFO).

Pile

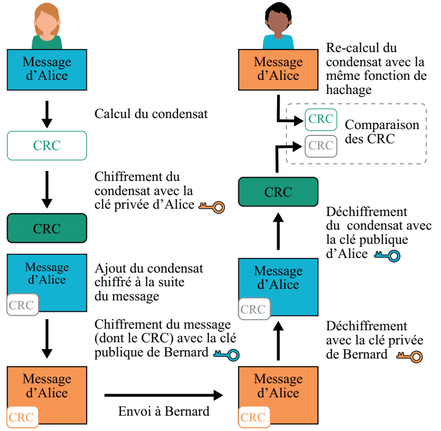


File



Sécurité

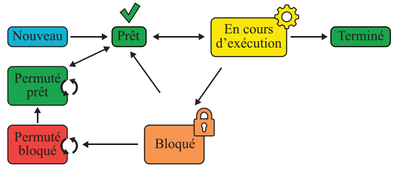
Principe de l'authentification d'un message entre deux correspondants :



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

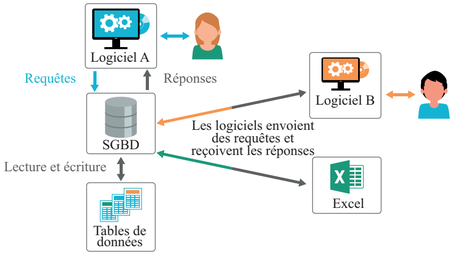


Deadlock (interblocage)

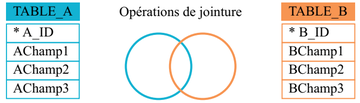
Deux processus (A) et (B) sont chacun en attente d'un travail que doit fournir l'autre.

Système de gestion de bases de données (SGBD)

Principe de fonctionnement d'un SGBD



Les différentes jointures de tables



Jointure gauche	`select * from TABLE_A as A left join TABLE_B as B on B.BChamp1=A.AChamp1`	
Jointure droite	`select * from TABLE_A as A right join TABLE_B as B on B.BChamp1=A.AChamp1`	
Intersection	`select * from TABLE_A as A inner join TABLE_B as B on B.BChamp1=A.AChamp1`	
Jointure gauche exclusive	`select * from TABLE_A as A left join TABLE_B as B on B.BChamp1=A.AChamp1 where B.BChamp1 is null`	
Jointure droite exclusive	`select * from TABLE_A as A right join TABLE_B as B on B.BChamp1=A.AChamp1 where A.AChamp1 is null`	
Jointure d'exclusion	`select * from TABLE_A as A full join TABLE_B as B on B.BChamp1=A.AChamp1 where A.AChamp1 is null or B.BChamp1 is null`