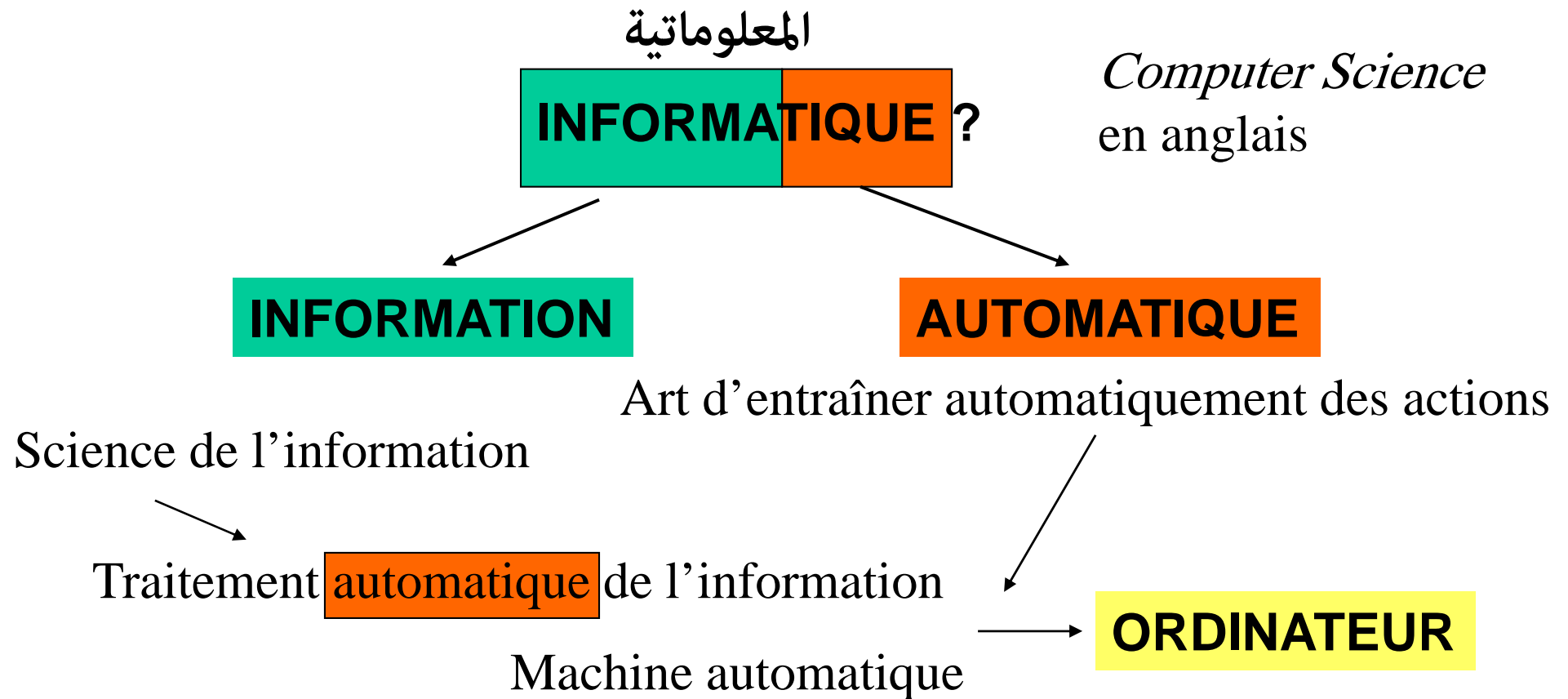


Cours d'Informatique
Initiation à l'informatique,
1^{ère} année LIC-GAT

Chapitre 1 : Introduction à l'informatique

1.1 : L'informatique ?



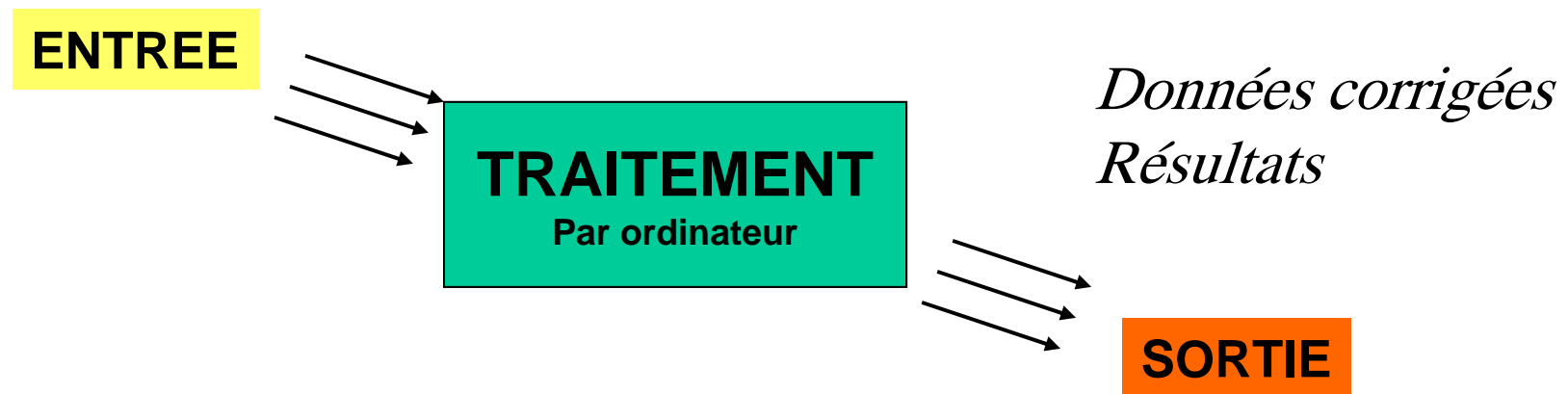
L'ordinateur

- Machine qui permet de traiter de l'information :
 - d'acquérir et de conserver de l'information (acquisition, stockage)
 - d'effectuer des traitements (calcul),
 - de restituer les informations stockées (restitution)
- Permet de lier «information» \Leftrightarrow «données» (0 ou 1)
- Différents types d'informations : valeurs numériques, textes, images, sons, ...: tout cela avec des 0 ou 1

Traitement de l'information

Schéma de principe du **traitement** de l'information

Données à l'état brut



L'ordinateur / l'homme

- Raison du remplacement :
 - Vitesse (pour des opérations « bas niveau »)
 - Fiabilité (répétitivité)
 - Mémoire
 - Coût
- 2 types d' « informaticiens »
 - les utilisateurs des outils informatiques
 - les concepteurs de ces outils : *votre but*

Domaines de l'informatique

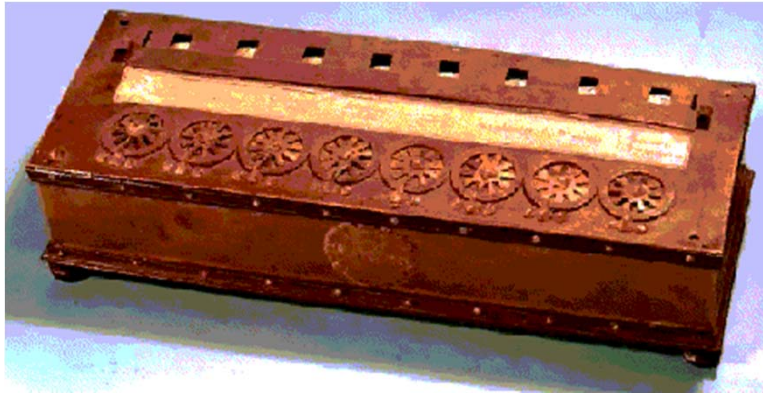
- Domaine du **matériel** (*hardware*)
 - partie physique de l'ordinateur
 - composants constituant un ordinateur (microprocesseur ...)
 - support du traitement de l'information (disque dur ...)
- Domaine du **logiciel** (*software*)
 - instructions expliquant à l'ordinateur comment traiter un problème
 - Cela nécessite de décrire des : algorithmes et représentations informatiques de ces instructions
 - Pour aboutir à un **programme**

1.2 : Matériel

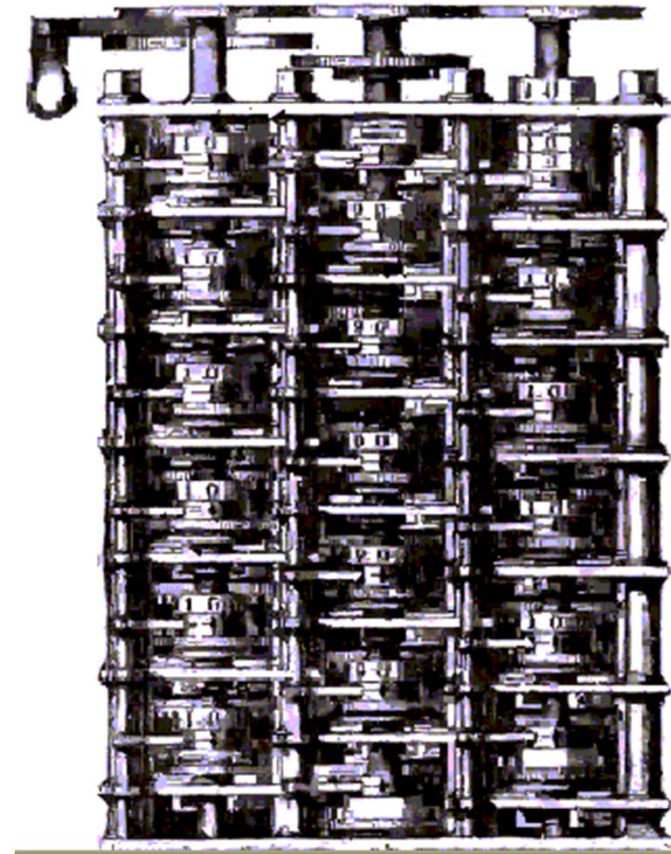
121 Aperçu historique

- **Principe de base** : John Von Newmann **1946** =véritable naissance de l'informatique
- Sinon les ancêtres et étapes importantes:
 - Boulier chinois
 - Numération binaire par Francis BACON en 1600
 - Machine à calculer de Pascal, 1642 : dépassée par l'apparition de l'électronique et des semi-conducteurs
 - Machine de Falcon, 1728. Première à utiliser un « programme » sur plaquette de bois perforée
 - Mémoire mécanique de Babbage, 1833. Conçoit une mémoire séparée des organes d'entrée et de sortie
 - Algèbre de Boole 1850 (bases de l'automatisme)
 - Machine de Hollerith, 1890. Il utilisa le premier la carte perforée, comme support universel d' information , mis en œuvre dans les premières générations d' ordinateurs.

Les ancêtres des ordinateurs



Machine de Pascal (1645)



1.2.2 - Générations d'ordinateurs

- Génération 1 (~1945 - 1960)
 - machines électroniques composées de circuits à lampes à vide (et non transistors à semi-conducteurs)
 - place importante (équivalent d'une salle)
 - performances de l'ordre de 1000 opérations/s
 - programmation en langage binaire
 - faible portabilité des programmes
 - programme et données fournis sous forme de cartes perforées, résultats sur une imprimante (pas de stockage)

1.2.2 - Générations d'ordinateurs

- Génération 2 (1960 - 1965)
 - découverte des transistors qui remplaceront les circuits à lampes à vide
 - Apparition des 1ère mémoires (à tores)
 - évite l'échauffement, gain de place, fiabilité
 - performances d'environ 100 000 opérations/s
 - programmation en langage binaire mais aussi à l'aide des **premiers langages évolués** (Fortran, Cobol, ...)

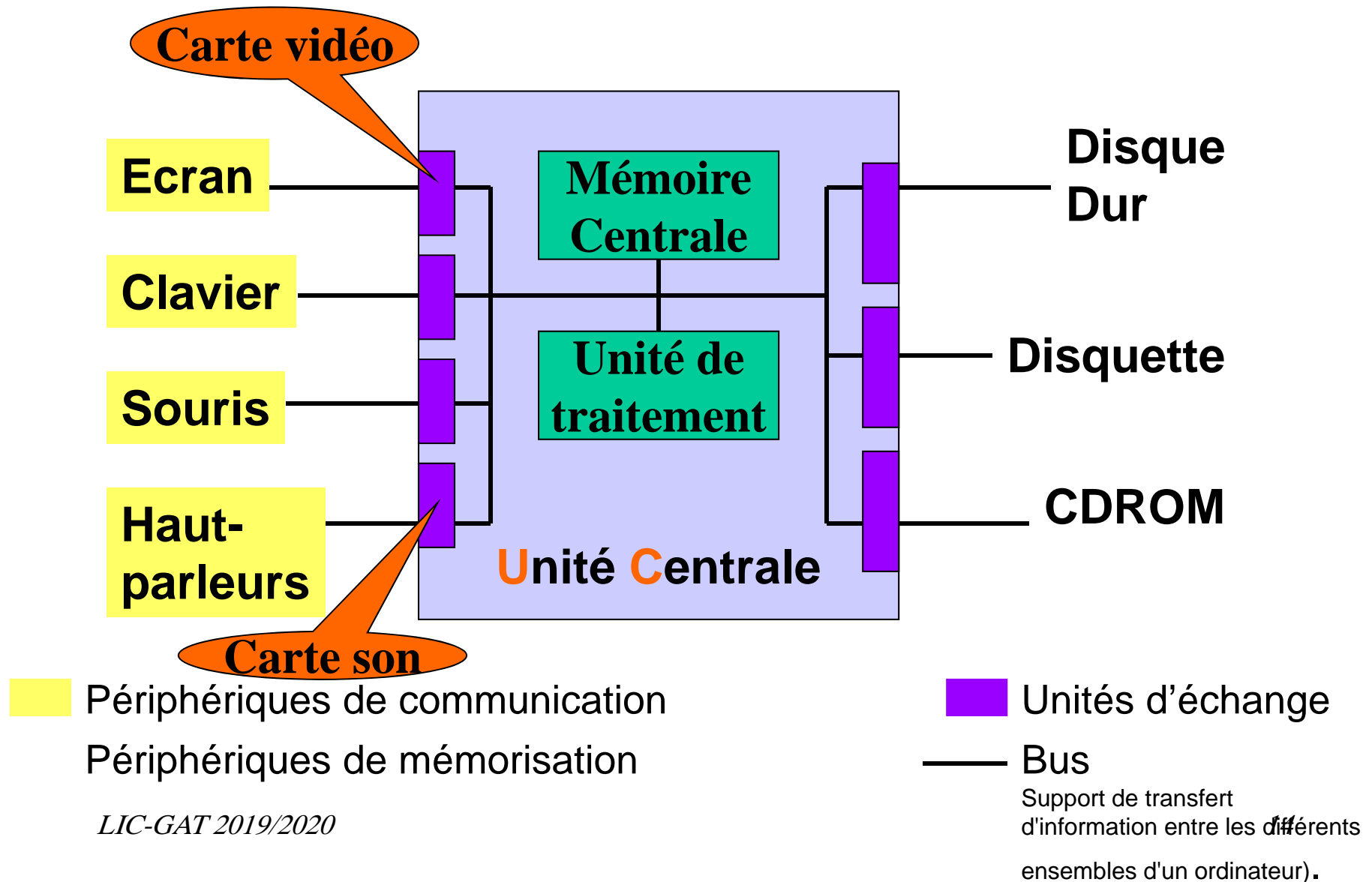
1.2.2 - Générations d'ordinateurs

- Génération 3 (1965 - 1975)
 - invention du circuit intégré permettant de placer des dizaines de transistors sur une puce de silicium
 - performances ↗ 10^9 à 10^{12} opérations/s
 - généralisation de la programmation en langage évolué
 - Les Systèmes d'Exploitation (**OS**) Permettent de gérer plusieurs programmes différents sous le contrôle d'un programme central

1.2.2 - Générations d'ordinateurs

- Génération 4 (1975 - ?)
 - exploitation du circuit intégré à grande échelle: plusieurs dizaines de milliers (millions) de circuits peuvent être intégrés sur une même puce
 - reproduction sur une seule puce d'une véritable micro machine : le micro processeur. (En 1971 l'Intel 4004 fut le premier microprocesseur)
 - diminution de la place occupé par un ordinateur
 - développement de l'**ordinateur personnel**.
 - La programmation s'oriente vers la **programmation OBJETS** (orientés autour des données et non plus des actions)

1.3 - Structure de l'ordinateur



1.3.1 - Périphériques

- 2 types de périphériques

Une pièce de matériel qui peut effectuer une fonction particulière

- périphériques de communication,
- périphériques de mémorisation.

- Périphériques de communication

- Périphériques d'entrées
 - clavier
 - souris
- Périphériques de sorties
 - écran
 - imprimantes

Les périphériques

ENTREES

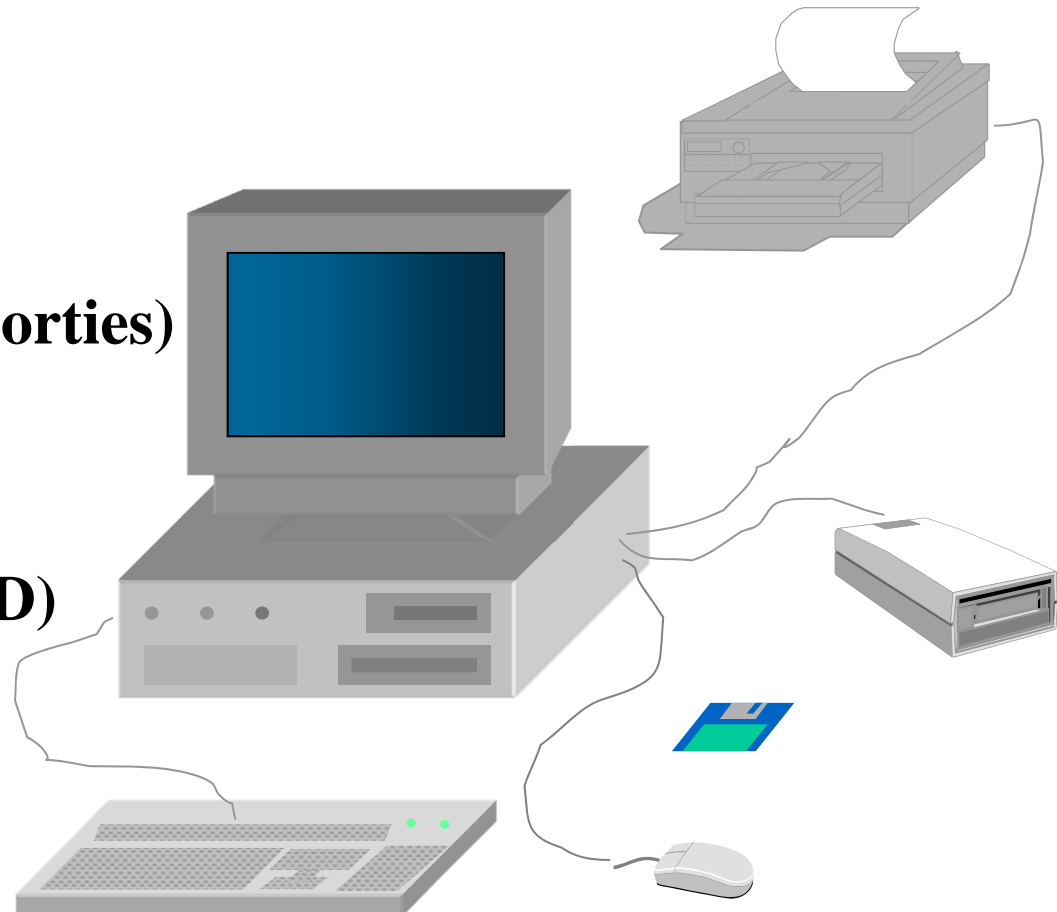
- Clavier
- Souris

MEMOIRES (entrées/sorties)

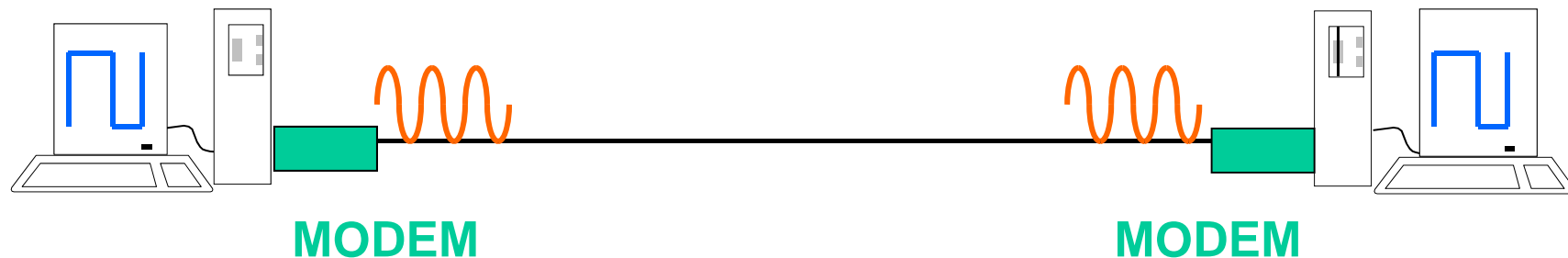
- Disque dur
- Disquette
- Disque optique (CD)

SORTIES

- Ecran
- Imprimante



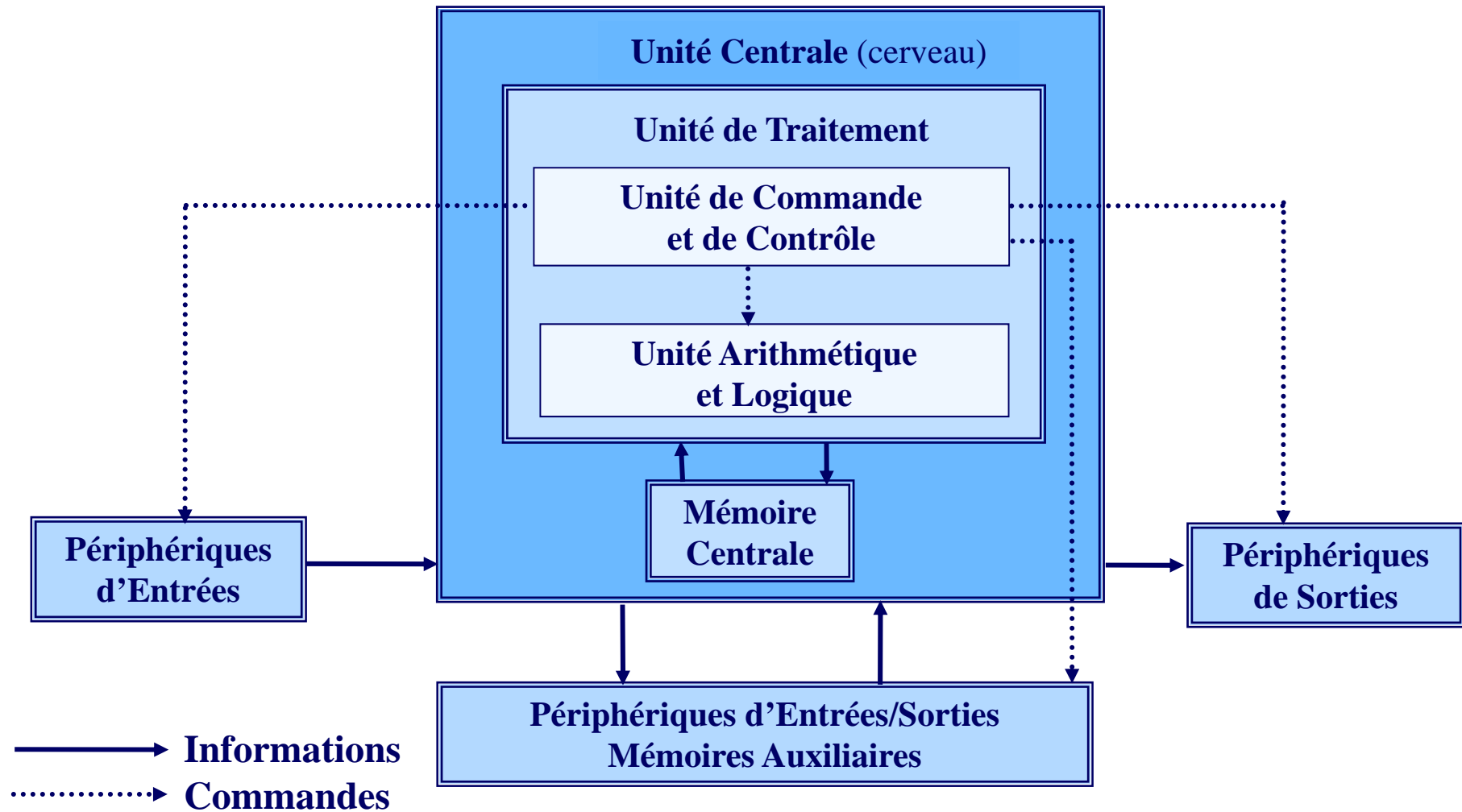
- Périphériques de communication
 - le modem
 - Permet : interconnexion des ordinateurs pour communiquer, échanger des informations, partager des applications
 - réseau d'ordinateur
 - utilisation du réseau téléphonique pour communiquer
 - ⇒ utilisation d'un **MODEM**



- Convertisseurs A/N ou N/A
- Signal numérique : 0 et 1
- Signal analogique: t et A continu => codage par modulation
- MODEM = modulation+démodulation

- Périphériques de mémorisation
 - permettent de sauvegarder et de restituer des informations
 - quantité d'informations pouvant être mémorisée se mesure en **Octet (8 éléments binaires)**
 - périphériques usuels de mémorisation:
 - disque dur
 - Disquette
 - CDROM, DVD

Schéma d'une configuration informatique



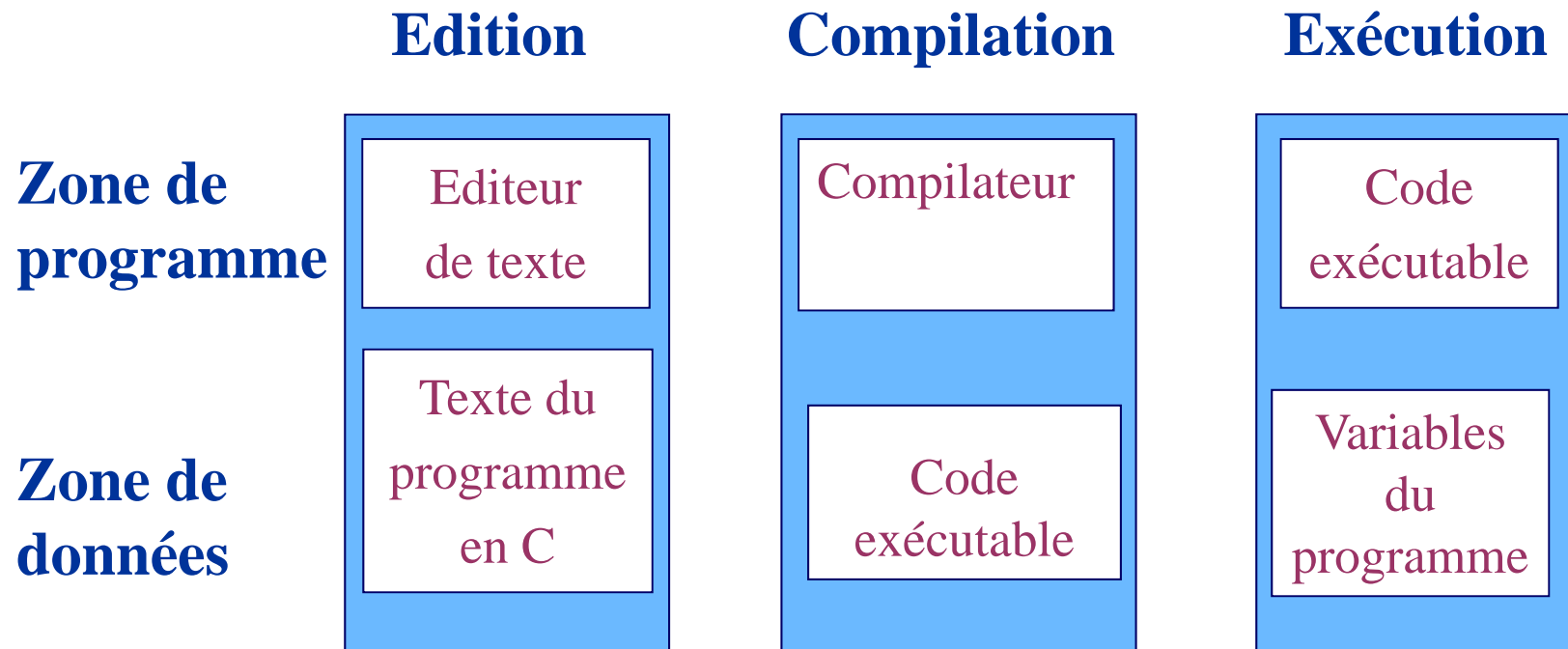
1.3.2 - La mémoire centrale

- Permet de stocker et de restituer des informations
- Correspond à un ensemble fini de **cellules**
- Une case mémoire (cellule) contient une donnée ou une instruction
- Chaque cellule est numérotée : **adresse**

a_1	a_2	a_3	a_4	a_5

1.3.2 - La mémoire centrale

**Contient le programme en cours d'exécution, ainsi que ses données.
Par exemple, lors de la création d'un programme en langage C:**



La mémoire auxiliaire

La mémoire centrale a une capacité limitée et est volatile

⇒ Nécessité d'enregistrer les informations sur des supports de mémoire non volatile (en sortie) pour pouvoir les réutiliser par la suite (en entrée).

FICHER : élément de mémoire de stockage (mémoire de masse) contenant de l'information et repérable.

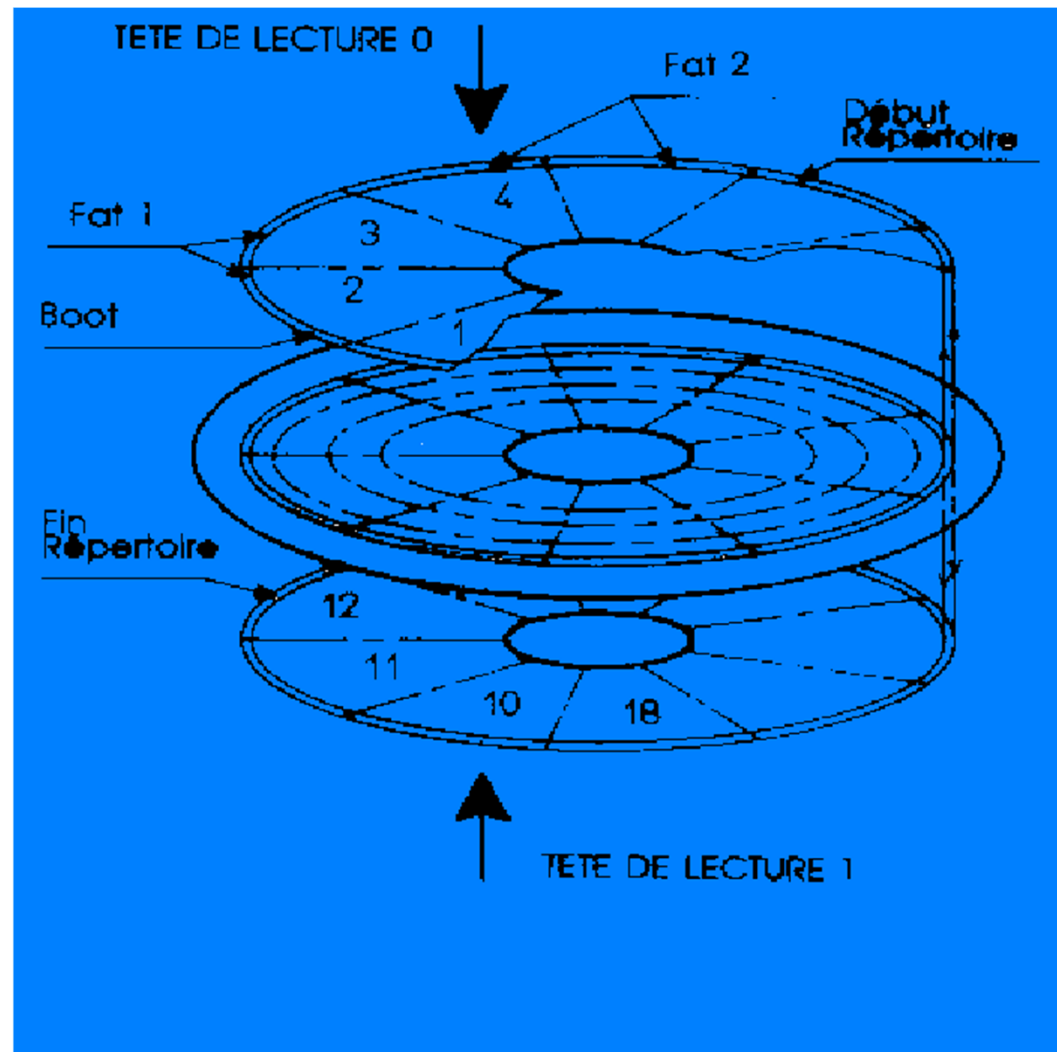
Le codage de l'information définit le **TYPE** du fichier et détermine son utilisation comme donnée pour un programme ou comme programme exécutable.

La mémoire auxiliaire

Les supports magnétiques à **ACCES ALEATOIRE** (on choisit la position que l'on veut) permettent une hiérarchisation des fichiers :

- ⇒ **REPERTOIRE** : fichier contenant le nom, l'adresse, ... de plusieurs autres fichiers
- ⇒ **RACINE** : fichier répertoire créé automatiquement sur le support au moment du formatage (disque ou disquette) et portant le nom du volume (label)
- ⇒ **CHEMIN D'ACCES** : liste des répertoires permettant d'atteindre un fichier à partir de la racine

La disquette



•Les faces

Comme tout disque, une disquette comporte deux faces utilisables pour y enregistrer l'information.

•Les pistes (tracks)

Les disquettes sont donc structurées en pistes concentriques sur lesquelles l'information sera écrite et lue. Les pistes qui se font face sur les deux côtés de la disquette et qui portent donc le même nom constituent un *cylindre*. Ceux-ci sont numérotés de la même façon que les pistes. Chaque piste est divisée en plusieurs morceaux: les secteurs.

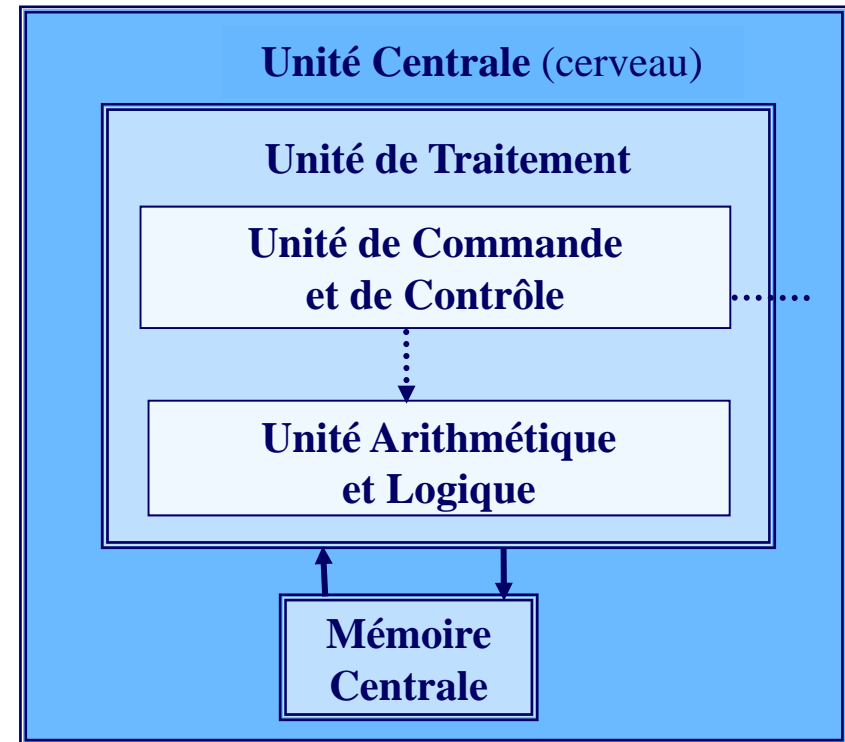
•Les secteurs (sectors)

De la même façon qu'une tarte est généralement découpée en morceaux de tailles égales, une disquette subit une découpe logique. Ainsi chaque piste est découpée en morceaux de tailles identiques appelés secteurs dont le nombre dépend du type de la disquette.

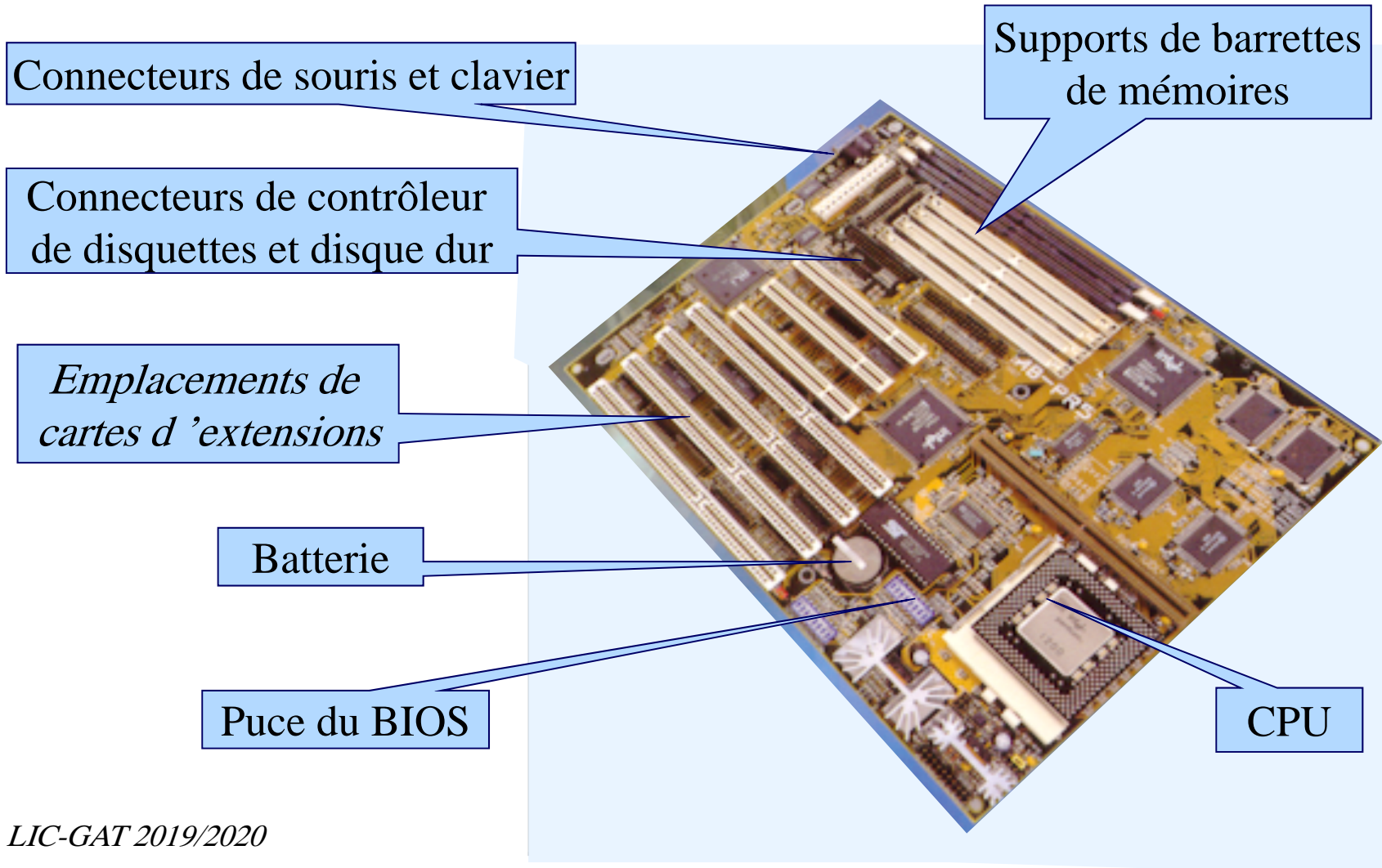
$$\text{Capacité} = \text{nombre de faces} * \text{nombre de pistes} * \text{nombre de secteurs} * \text{taille d'un secteur}$$

1.3.3 - L'unité de traitement

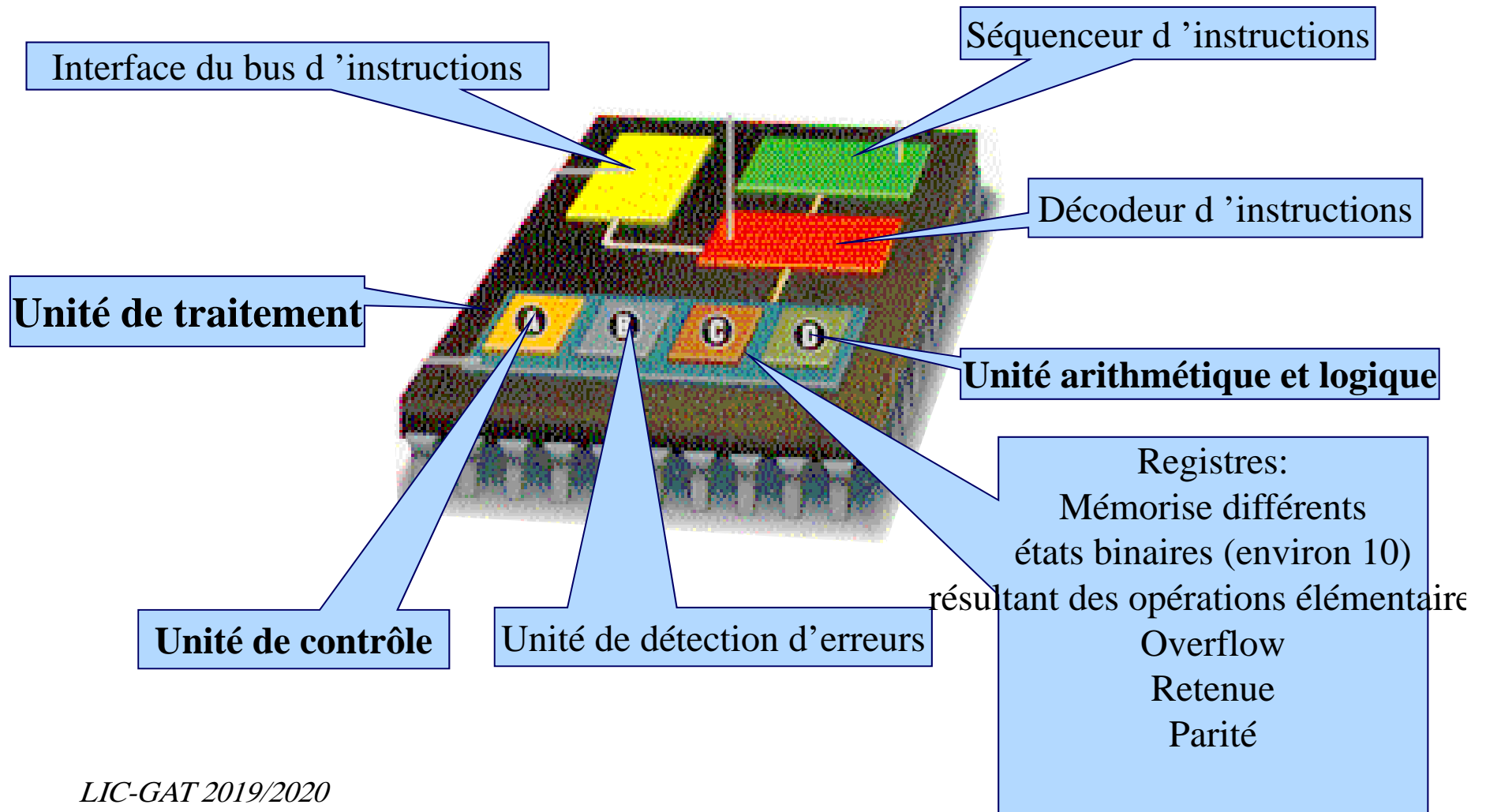
- **Unité de commandes**
 - unité de contrôle
 - chef d'orchestre
 - conduit le calcul
(ordre des instructions)
voulu par le programmeur
- **Unité Arithmétique et logique**
 - unité de calcul
 - exécute les opérations arithmétiques et logiques (+-*/ et ou ...) d'un programme sous le contrôle de l'unité de commande



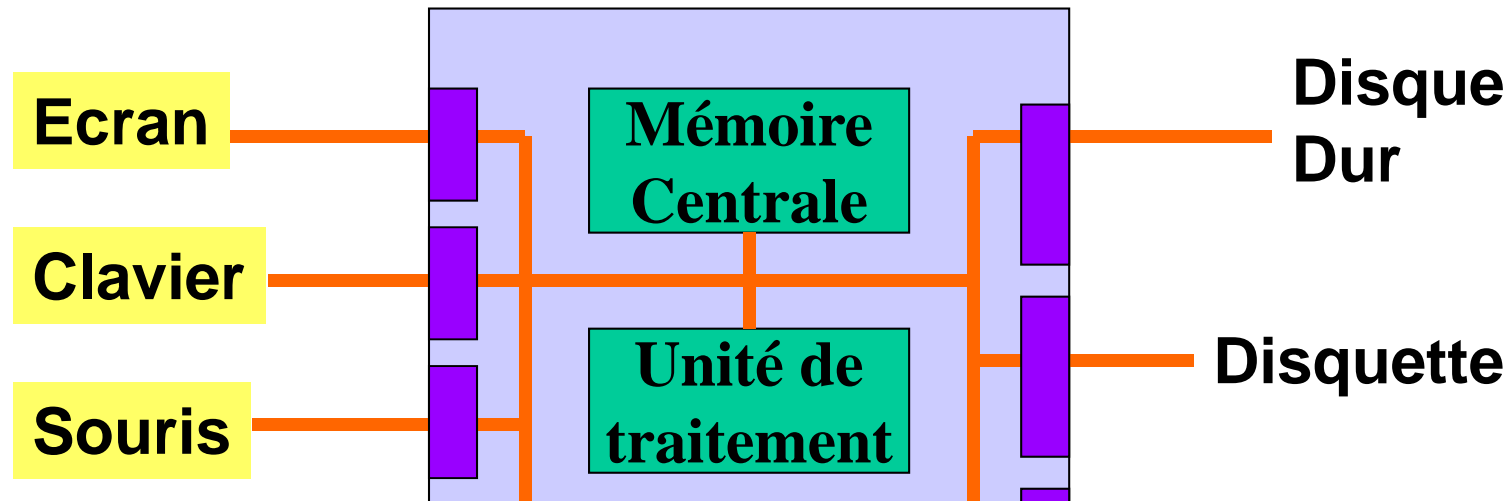
L'unité centrale CI sur lequel on trouve



Le processeur (CPU)



1.3.4 - Les bus



- Permettent de faire le lien entre les différentes unités d'un ordinateur
- représente le chemin utilisé par les informations pour aller d'une unité à l'autre

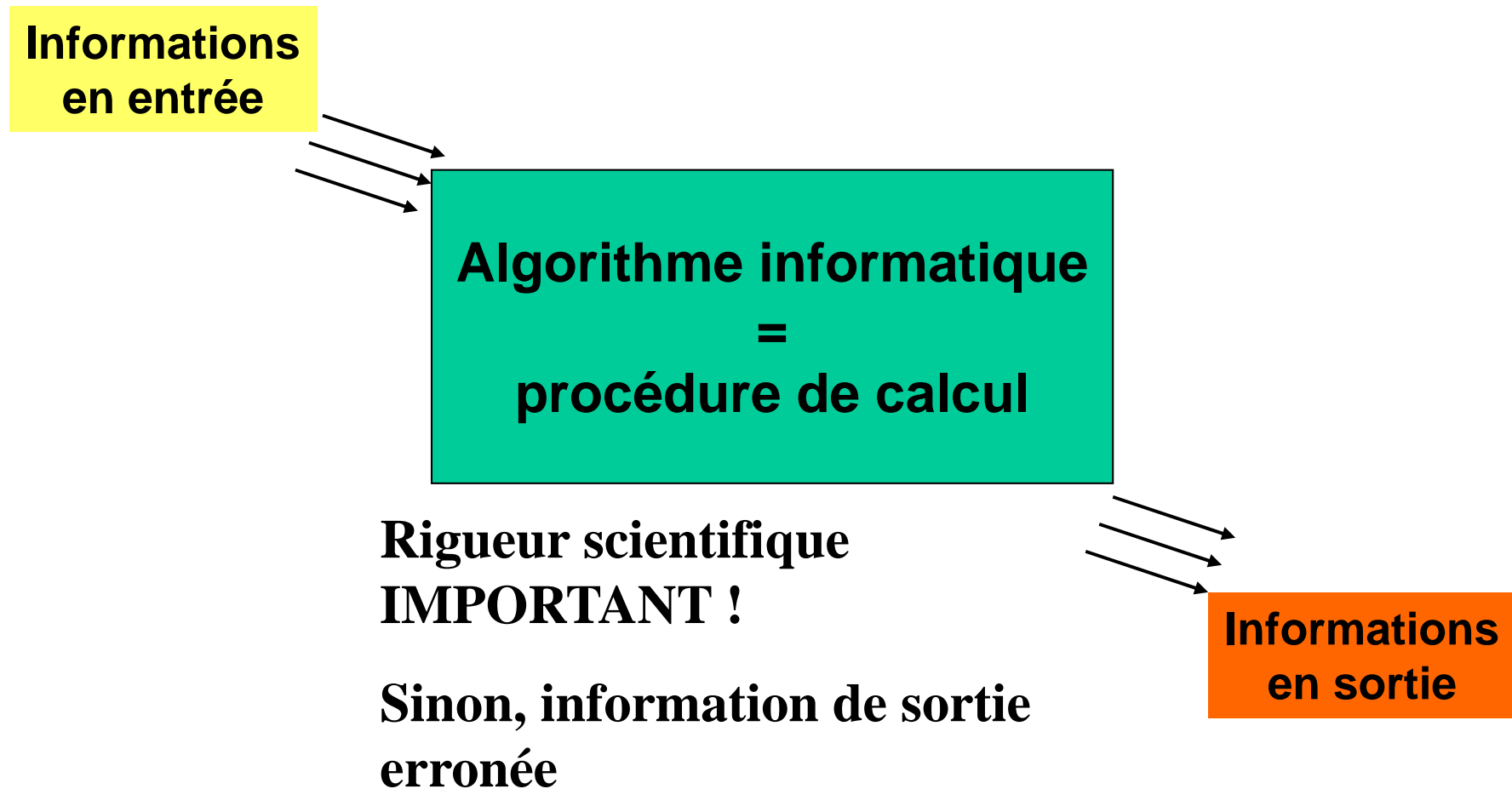
1.4 : Logiciel

- Domaine du **logiciel** (*software*)
 - instructions expliquant à l'ordinateur comment traiter un problème
 - algorithmes et représentation informatiques de ces instructions
 - **programme**

1.4.1 - Algorithmes et programmes

- Algorithme :
 - suite des actions à effectuer pour
 - réaliser un traitement donné
 - résoudre un problème donné
- Exemples d'algorithme dans la vie courante
 - pour tricoter un pull : (maille à l'endroit, ...)
 - pour faire la cuisine : recette
 - pour jouer une sonate : partition

1.4.1 - Algorithmes et programmes



1.4.1 - Algorithmes et programmes

- Programme :
 - codage d'un algorithme afin que l'ordinateur puisse exécuter les actions décrites
 - doit être écrit dans un langage **compréhensible** par l'ordinateur
 - → langage de programmation (Assembleur (micropro), Basic, C, Fortran, Pascal, Cobol ...)
- Un programme est donc une suite ordonnée d'instructions élémentaires codifiées dans un **langage** de programmation

1.4.2 - Langages de programmation

- L'ordinateur
 - construit autour d'un ensemble de circuits électroniques (le courant passe, le courant ne passe pas)
 - traite donc que des signaux assimilables à 0 ou 1
 - une opération élémentaire \rightarrow suite de 0 et de 1 = suite de bits (BInary digiT) ! Un champ de 8 bits constituant ce qu'on appelle 1 byte ou 1 octet. Importance des unités en science. Rappel: $k(2^{10})$ M et G .
- Pour que les programmes et les données soient compréhensibles par l'ordinateur il faut effectuer un codage binaire

1.4.2 - Langages de programmation

- Langage machine
 - langage binaire
 - ses opérations sont directement compréhensibles par l'ordinateur
 - **propre à chaque famille d'ordinateur**
 - *Pour pouvoir manipuler du **langage machine**, on est obligé de passer par de l'**Assembleur**.*
- Ecriture des premiers programme en langage machine

Les langages

Rédiger un programme consiste à préparer le travail à **FAIRE FAIRE** à la machine, sous forme d'une liste d'instructions.

Les instructions que peut exécuter l'unité de traitement sont codées en **langage binaire** spécifique à chaque machine (langage machine).

Langage du programmeur → langage machine (code binaire exécutable).
La traduction est effectuée par un **compilateur** (qui est lui aussi un programme ...).

TYPES DE LANGAGES → STYLE DE PROGRAMMATION

- langages impératifs (Fortran, Pascal, C ...) : Il s'agit de faire exécuter une suite d'ordres par une machine bête mais disciplinée.
- langages Déclaratifs: l'activité de programmation consiste essentiellement à décrire le **rapport** qui existe entre les données et les résultats que l'on veut obtenir, plutôt que la séquence de traitements qui mène des uns aux autres
 - fonctionnels (Lisp, Scheme ...)
 - logiques (Prolog ...)
- langages objets (C++, VisualBasic, Delphi, Java ...)

- Habituellement, un programme c'est une suite d'instructions. L'ordinateur est très bête et il faut tout lui détailler:
 - Mettre la main sur la poignée de la porte
Tourner la poignée
Pousser la porte
Mettre le doigt sur l'interrupteur
Appuyer sur l'interrupteur pour allumer l'ampouleLe
programme principal
- Tout se passe très bien. Mais qu'est-ce qui se passe par exemple si on met une porte automatique ? Le programme sera incapable de trouver la poignée et d'ouvrir la porte !
- En programmation objet, on associe aux objets des actions (aussi appelées *méthodes*). Par exemple, à l'objet **porte** on peut associer la méthode **ouvrir**. De même pour l'ampoule on pourrait associer une méthode **allumer**, **éteindre**, etc.
- Le programme devient plus simple:
 - porte.ouvrir
ampoule.allumer
» Le nouveau programme principal
- On a plus besoin de savoir comment la porte s'ouvre. On se contente de l'ouvrir.
- Pour indiquer qu'on applique la méthode (ouvrir) sur l'objet (porte), on note souvent **objet.méthode** (ici : porte.ouvrir).

- Bien sûr il faut détailler ce que fait la méthode **ouvrir** de **porte** et la méthode **allumer** de **lumière**. On ne va pas détailler dans le programme ce que fait la méthode ouvrir, mais on va le détailler **dans l'objet lui-même**.
- C'est normal, puisque la méthode ouvrir ne s'applique qu'à la porte, pas à la lumière:
 - `porte.ouvrir:`
 - Mettre la main sur la poignée
 - Tourner la poignée
 - Pousser la porte
- **L'objet *porte* et sa méthode**
- On peut changer la porte en mettant une porte automatique. On peut aussi l'ouvrir (même si la porte elle-même ne s'ouvre pas de la même façon):
 - `porte.ouvrir:` Se placer devant la porte
 - Attendre que la porte soit complètement ouverte
 - **L'objet *porte automatique* et sa méthode**
- Mais votre programme pourra l'ouvrir sans rien changer:
 - `porte.ouvrir`
 - `ampoule.allumer`
- **Le programme principal : il est inchangé malgré le changement de porte**

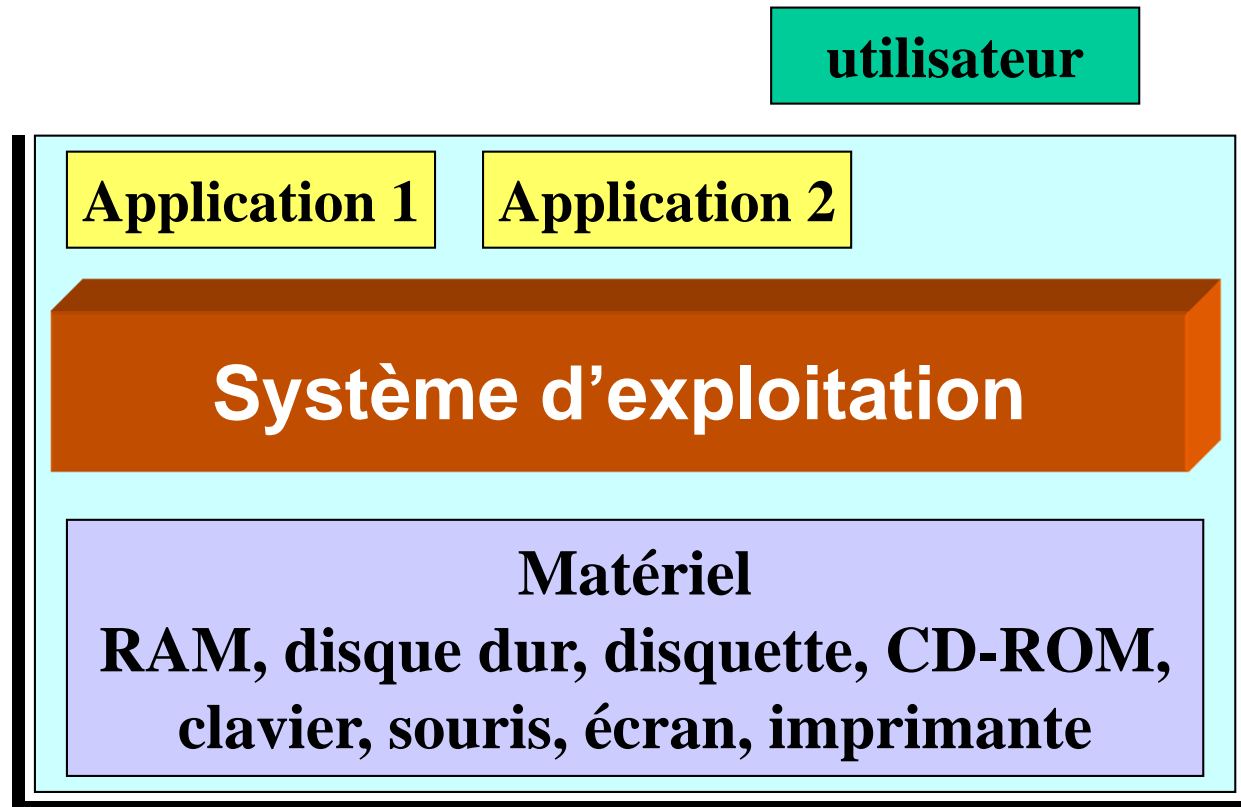
1.5 : Système d'exploitation

- Tout premier logiciel indispensable au bon fonctionnement de l'ordinateur
- Chargé de la gestion globale de l'ordinateur notamment :
 - utilisation de la mémoire vive
 - utilisation de la mémoire auxiliaire
 - dialogue avec l'utilisateur (*i.e.* les entrée/sorties)

Systeme d'exploitation

- Gestion de la mémoire vive
 - aller chercher sur la mémoire auxiliaire le code (exécutable) et les données nécessaires à l'exécution d'un programme demandé par l'utilisateur
- Gestion des entrées/sorties
 - interprétation des frappes de touches au clavier et des mouvements ou cliques de la souris
 - visualisation d'informations à l'écran

Système d'exploitation



Systeme d'exploitation

- Les plus courants :
 - système mono utilisateur
 - MS-DOS pour PC
 - Windows 95-98 pour PC
 - MAC-OS pour Macintosh
 - système multi utilisateurs
 - OS/2 pour PC (IBM)
 - Windows NT pour PC
 - UNIX

– **Linux sur PC (gratuit)** ; Solaris sur SUN ; AIX sur IBM

3 types d'ordinateurs:

- **PC=perso** (plusieurs marques compatibles)

- **Mac=perso**

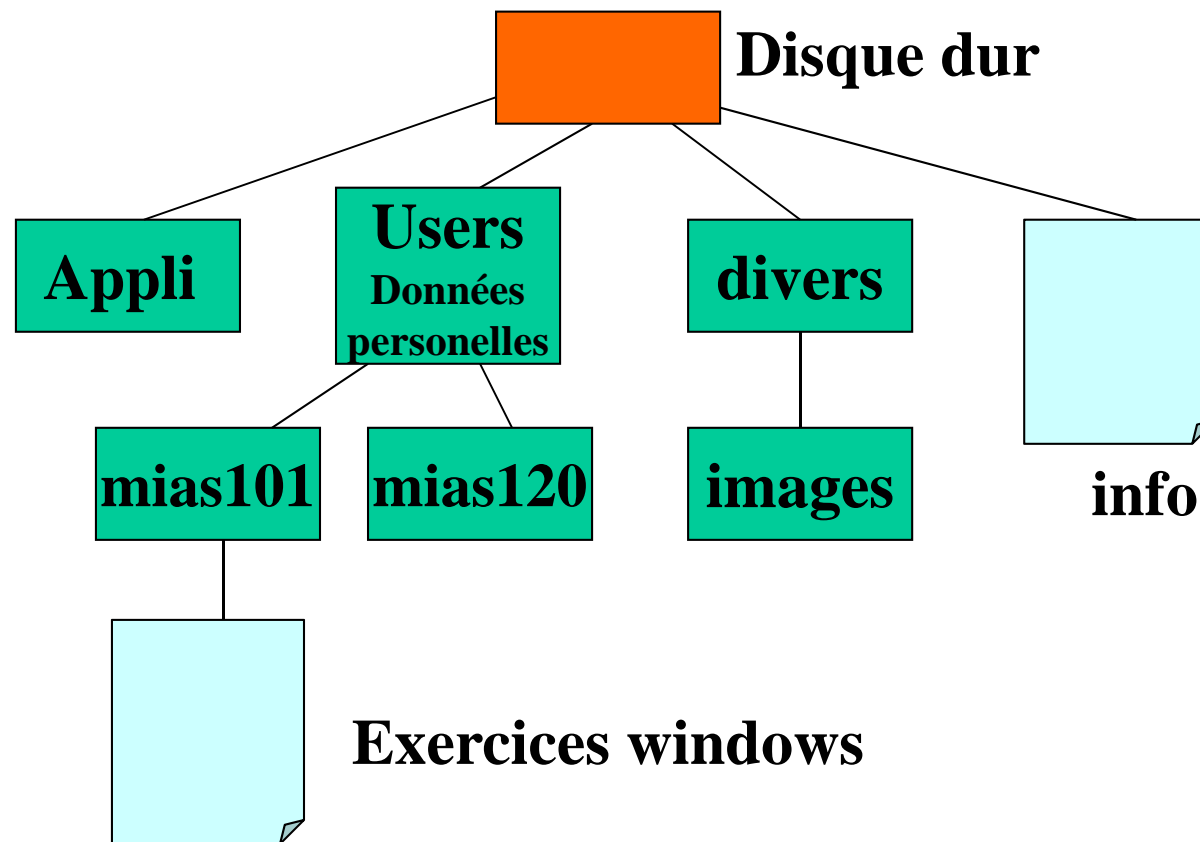
- **Stations de travail=professionnel**

! compatibilité

Systeme d'exploitation

- Permet aussi d'organiser (gérer) les informations sur la mémoire auxiliaire
 - **FICHIER** (*file*): ensemble d'information manipulables par l'ordinateur ; nom unique dans un répertoire ; taille en **octets**
 - **REPERTOIRE** : dossier ou *directory* ; regroupement de fichiers ; nom unique
 - **RACINE** (*root*): dossier situé au plus haut niveau
 - **CHEMIN D'ACCES** (*path*): liste des répertoires permettant d'atteindre un fichier à partir de la racine

Systeme d'exploitation



1.6 : Les applications

Une application est un logiciel fournissant à l'utilisateur un environnement (fonctionnalités) lui permettant de créer, modifier et/ou consulter des **documents** contenant des informations.

Applications et documents

- ⌘ Interface (menus, barres d'outils, ...)
- ⌘ Fonctions de gestion des documents (ouvrir, enregistrer, ...)
- ⌘ Fonctions d'édition (couper, copier, coller, ...)
- ⌘ Fonctions de recherche et remplacement
- ⌘ Insertions d'images, objets ... (incorporations...)
- ⌘ Outils de mise en forme (polices de caractères, styles ...)
- ⌘

Editeur de texte

Logiciel permettant de créer des documents contenant du texte au codage ASCII

(C'est une norme de codage de 128 caractères alphanumériques sur 7 bits. Les versions étendues sur 8 bits, 256 caractères sont adaptées suivant les pays.)

•

Outil du système d'exploitation (fichiers de configuration, notices, ...)

Bloc-Notes

Traitement de texte

Logiciel permettant de créer des documents contenant du texte mis en forme : police et styles de caractères, style et alignement des paragraphes, organisation du texte (sections, hiérarchisation, en-tête et pied de page, table des matières ...) etc...

WordPad

Word

Traitement d'image

Logiciel permettant de créer des images définies sous forme de matrice de points de couleurs (Pixels).

Inconvénient: ligne=escalier

Paint

Dessin vectoriel

A la différence d'une image définie point par point, un élément de dessin vectoriel se définit par sa forme, sa position et sa taille.

Un dessin se compose de plusieurs formes (rectangles, cercles, traits, ...) et chaque forme est modifiable.

Pas l'inconvénient du transparent précédent

Dessin

Tableur

Logiciel permettant de créer des documents contenant des feuilles de calcul constituées de cellules, organisées en ligne x colonne, pouvant contenir des informations sous différents formats (texte, nombres, dates ...) ou des formules de calcul.

Contient des outils d'analyse de ces informations permettant une visualisation sous forme de courbes ou diagrammes divers.

Excel

Système de Gestion de Bases de Données

Logiciel permettant de créer des documents contenant des tables d'articles hiérarchisées ou liées par des champs communs (relations).

Ex: Généalogie. Listes d'étudiants, avec adresse, téléphone, niveau, ...

Contient des outils d'analyse des tables permettant d'en extraire des informations à l'aide de requêtes.

Ex: Combien d'étudiants de la fac habitent à Sainte Rose et sont en Deug Mass

[Access](#)

Applications multimédia

Logiciels permettant de reproduire ou de créer des sons, des animations, des vidéo, ...

Navigateur

Logiciel permettant de visualiser des pages écrites en langage HTML et disponibles sur le réseau Internet.

[Netscape](#)

Présentations

Il existe aussi des logiciels de présentation permettant d'afficher une suite de pages à l'écran sous forme de diaporama.

PowerPoint