

Cours d'Informatique  
*Initiation à l'informatique et  
à l'algorithmique*  
1<sup>ère</sup> année

- **Semestre 1**
- **UE1.4 : Algorithmie**
- **Coef: 03**

# Présentation des étudiants

- Nom et prénoms :
- Série BAC:
- Expériences en Algorithmie
- Vos attentes de la formation

# Conditions générales de travail

- **Volume horaire : 40h ( $CM=15h$   $TD = 25h$ )**
- **Assiduité et ponctualité, travail de groupe recommandé**
- **Consignes sur les téléphones portables**
- **Etat de veille/somnolence**
- **Réalisation des devoirs (Un (01) contrôle + un (01) sujet d'examen)**
- **Fraudes et tentatives de fraudes**

# Objectifs pédagogiques du cours

## Objectif général

**A la fin du cours, l'étudiant est capable d'écrire un algorithme simple**

# Objectifs pédagogiques du cours

## Objectifs spécifiques (1/2)

**A la fin du cours, l'étudiant est capable :**

- **de comprendre ce qu'est un algorithme**
- **de comprendre ce qu'est un programme**
- **de comprendre le processus de réalisation d'un programme (cycle de développement du logiciel)**
- **d'utiliser des variables simples dans des expressions de calcul et d'affectation**

# Objectifs pédagogiques du cours

## Objectifs spécifiques (2/2)

**A la fin du cours, l'étudiant est capable :**

- **d'utiliser des structures de contrôle**
- **d'utiliser des structures de données de base**
- **d'écrire et d'utiliser des procédures, des fonctions et les modes de passage de leurs paramètres**
- **d'utiliser la récursivité lors de l'écriture de fonctions**

# Contenu du cours

- **Chapitre 0 – Introduction à l'informatique**
- **Chapitre 1 – Introduction à l'algorithmique**
- **Chapitre 2 – Variables, constantes et instructions de base en algorithmique**
- **Chapitre 3 – Structures de contrôle en algorithmique**
- **Chapitre 4 – Structures de données de base en algorithmique**
- **Chapitre 5 – Procédures, fonctions et les modes de passage de leurs paramètres**
- **Chapitre 6 – Fonctions récursives**

# Bibliographie

- **Introduction à l'algorithmique, 2004, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest et Clifford Stein, Dunod, Paris ;**
- **Wirth N., Algorithmes et structures de données, Eyrolles, Paris, 1989 ;**
- **Biondi J., Clavel G., Introduction à la programmation, algorithmique et langage , Masson, Paris, 1987.**



# Chapitre 0 : Introduction à l'informatique

Chapitre suivant algorithmique

# 1.1 : L'informatique ?

**INFORMATIQUE ?**

*Computer Science*  
en anglais

**INFORMATION**

**AUTOMATIQUE**

Art d'entraîner automatiquement des actions

Science de l'information

Traitement **automatique** de l'information

Machine automatique

**ORDINATEUR**

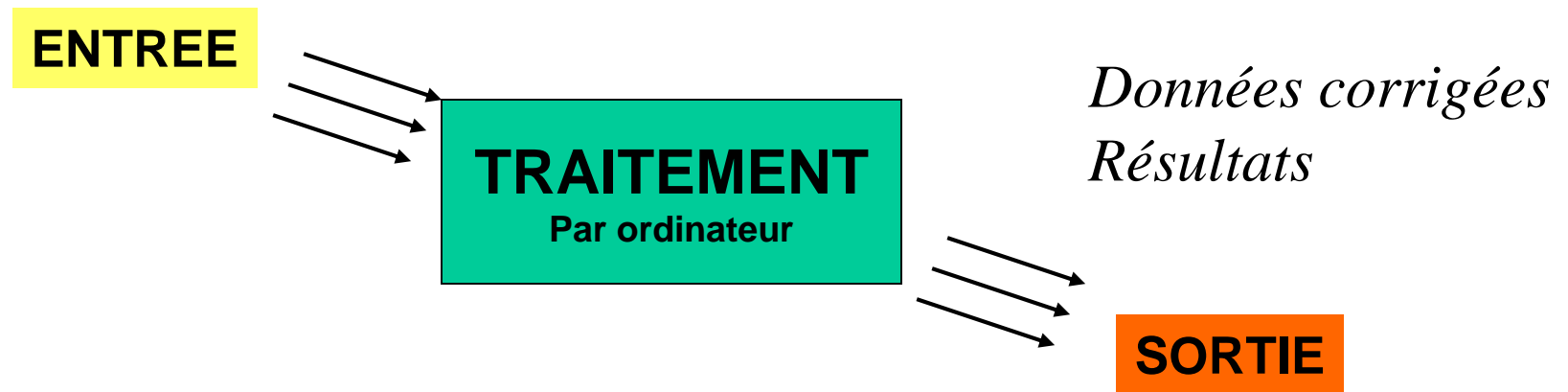
# L'ordinateur

- 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

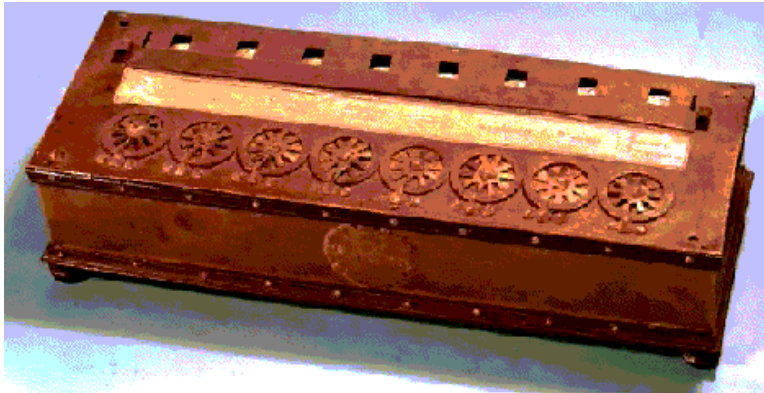
- Domaine du **matériel** (*hardware*)
  - partie physique de l'ordinateur
  - composants constituant un ordinateur (microprocesseur ...)
  - support du traitement de l'information (disque dur ...)
- Domaine du **logiciel** (*software*)
  - instructions expliquant à l'ordinateur comment traiter un problème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

# 1.2 : Matériel

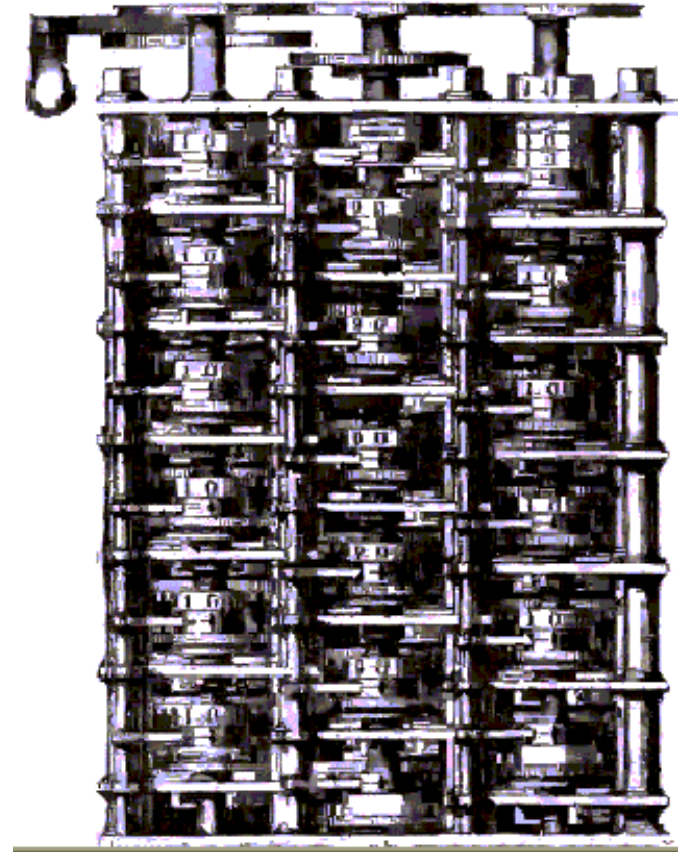
## 121 Aperçu historique

- **Principe de base** : John Von Neumann **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 oeuvre 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, ...)

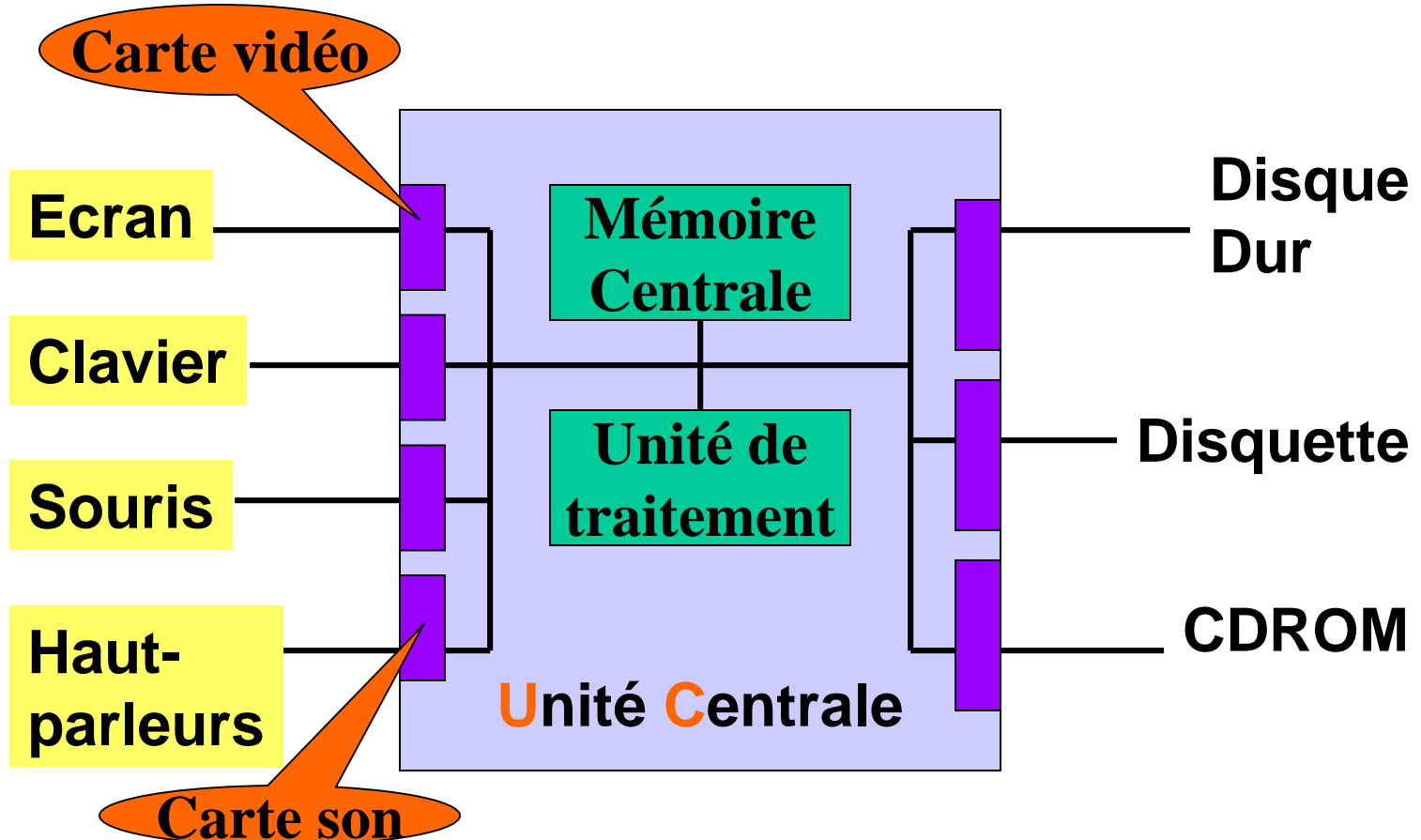
## 1.2.2 - Générations d'ordinateurs

- 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



■ Périphériques de communication

■ Périphériques de mémorisation

■ Unités d'échange

— Bus

Support de transfert  
d'information entre les différents  
ensembles d'un 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

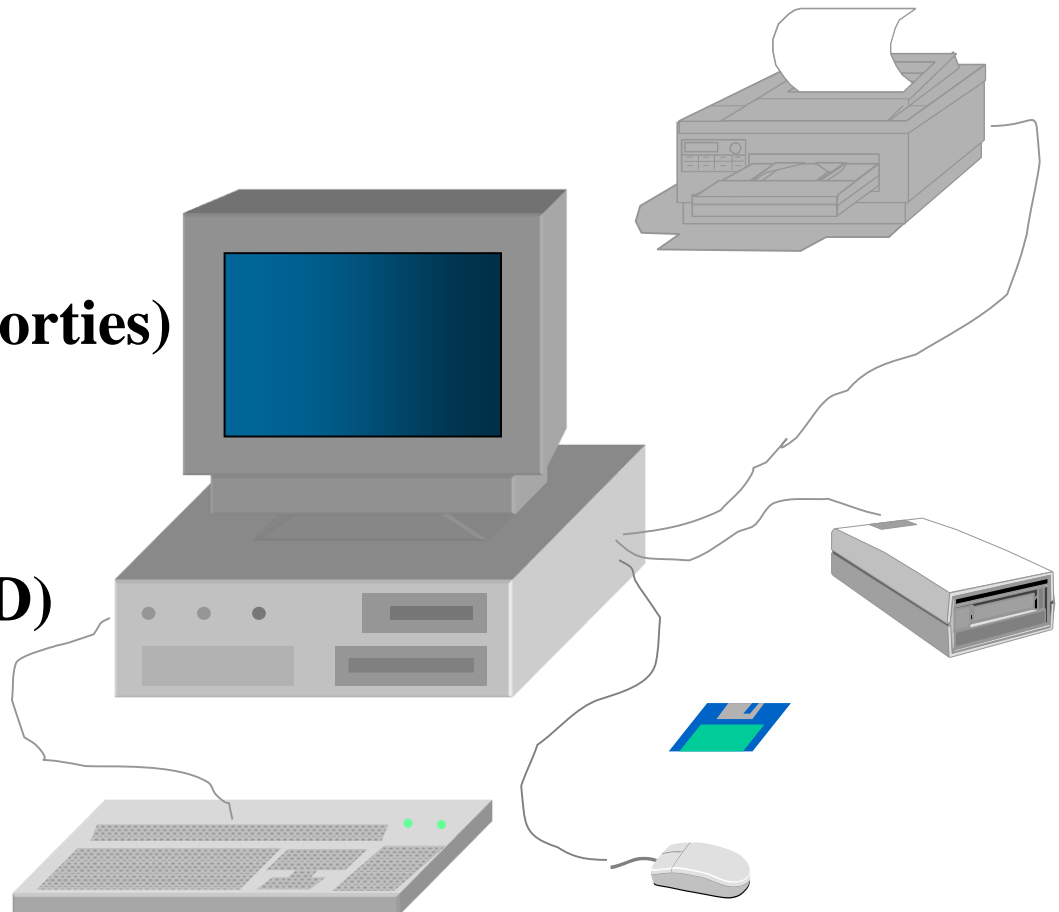
- Clavier
- Souris

## MEMOIRES (entrées/sorties)

- Disque dur
- Disquette
- Disque optique (CD)

## SORTIES

- Ecran
- Imprimante



## 1.4 : Logiciel

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

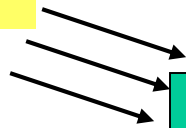


## 1.4.1 - Algorithmes et programmes

- 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

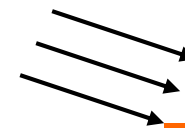
**Informations  
en entrée**



**Algorithme informatique  
=  
procédure de calcul**

**Rigueur scientifique  
IMPORTANT !**

**Sinon, information de sortie  
erronée**



**Informations  
en sortie**

## 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 → suite de 0 et de 1 = suite de bits (BInary digiT) !.
- 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 unes aux autres
  - fonctionnels (Lisp, Scheme ...)
  - logiques (Prolog ...)
- langages objets (C++, VisualBasic, Delphi, Java ...)

## 1.5 : Système d'exploitation

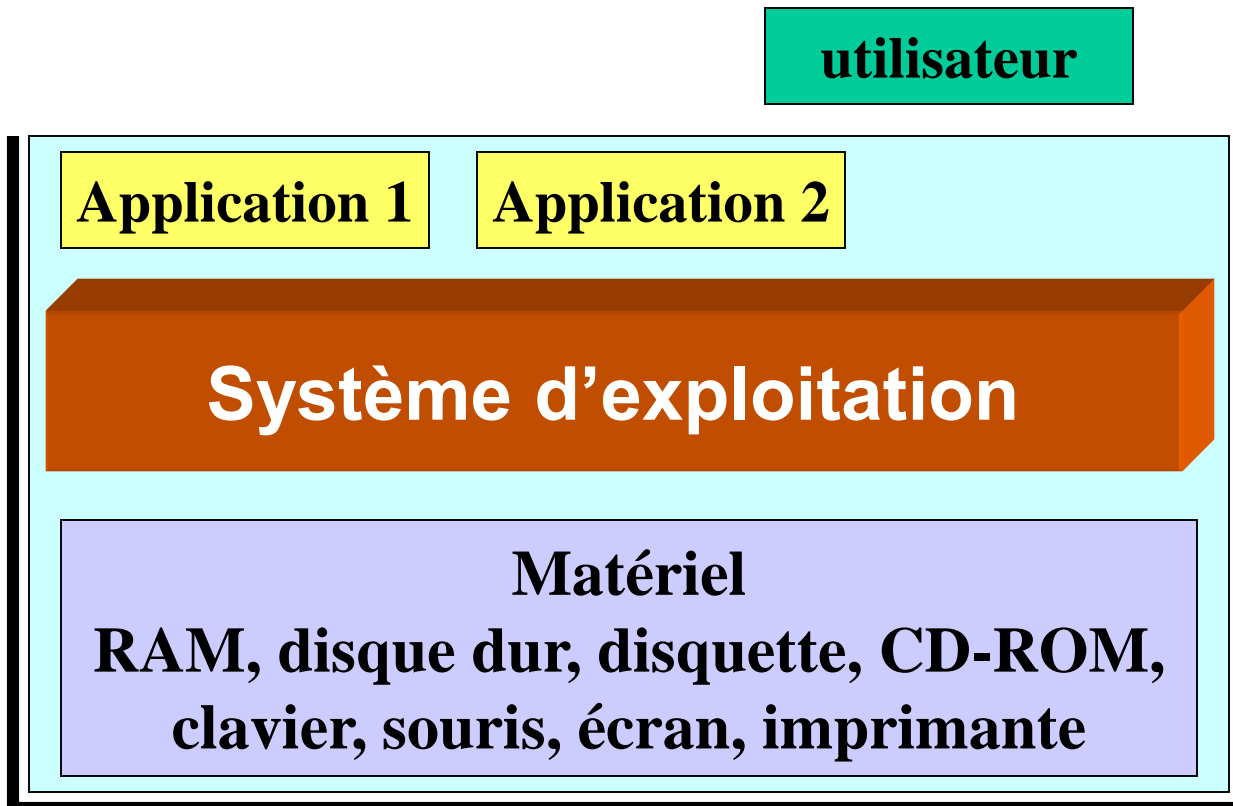
- Tout premier logiciel indispensable au bon fonctionnement de l'ordinateur
- Chargé de la gestion globale de l'ordinateur notamment :
  - utilisation de la mémoire vive
  - utilisation de la mémoire auxiliaire
  - dialogue avec l'utilisateur (*i.e.* les entrée/sorties)

# Système 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



# Systeme 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

## 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

# Systeme 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

Fin