



الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة الجبلاي بونعامة خميس مليانة

Université Djilali BOUNAAMA Khemis Miliana

كلية العلوم الطبيعية والحياة وعلوم الأرض

Faculté : Sciences de la Nature, de la Vie et des Sciences de la Terre

## Polycopié de cours

**Intitulé :**

Système D'exploitation, Les Logiciels Et Algorithmique

**Module :**

**Outils Informatique**

**Cours destiné aux étudiants de la deuxième année licence du domaine**

**Sciences de la nature et de la vie**

**Départements :**

Sciences Biologiques

**Filières :**

- Sciences Biologiques
- Biotechnologie

**Présenté par**

**Dr. Baouche Fatima Zohra**

**UDBKM 2021/2022**

## **Avant-propos**

Ce support de cours est destiné aux étudiants de la deuxième année licence des deux filières « Sciences Biologiques » et « Biotechnologie » du domaine « sciences de la nature et de la vie ». Le but est de faire apprendre aux étudiants l'ensemble des mécanismes et outils nécessaires à la mise en œuvre d'un système d'exploitation qui est très important pour permettre l'adaptation de la machine aux besoins réels des utilisateurs. La deuxième partie est consacrée à une présentation générale de la méthode d'écriture des algorithmes scientifiques afin de comprendre le fonctionnement des logiciels d'un ordinateur. Le cours pourra être destiné aussi aux étudiants de la deuxième année licence des deux filières: « Ecologie Et Environnement » et « Sciences Hydro Biologique Marine Et Continentale ».

Le polycopié de cours est un ensemble de connaissances théoriques confectionnées de plusieurs références bibliographiques citées à la fin de chaque chapitre. Les informations collectées sont jugées nécessaires pour aborder des applications concrètes permettant aux étudiants de comprendre leurs cours afin de les initier à utiliser les ordinateurs et les programmes informatiques dans leurs cursus.

Le cours se compose de deux chapitres d'importances égales. Le premier chapitre est basé sur l'éclaircissement du fonctionnement des systèmes d'exploitation Windows, Unix et Linux en présentant alors ce qui caractérise ce système en générale, ses différents types, leur rôle et leurs modes d'utilisation. Dans le deuxième chapitre, nous décrivons les principaux types des logiciels et les instructions nécessaires utilisées par langage algorithmique pour aider les étudiants à écrire par la suite des programmes informatique.

**L'auteur**

**Dr. Baouche Fatima Zohra**

## CHAPITRE I : DECOUVERTE DU SYSTEME D'EXPLOITATION

### SOMMAIRE

I.1 Introduction .....	2
I.2 Évolution des versions de Windows.....	2
I.3 Système de gestion de Windows.....	5
I.3.2 Le poste de travail .....	8
I.3.3 La fenêtre d'une application (d'un programme) sous Windows.....	10
I.3.4 Explorateur Windows.....	13
I.3.6 Rechercher des fichiers sous Windows .....	14
I.3.7 Panneau de configuration .....	15
I.4 Opérations de gestion sous Windows .....	17
I.4.1 Propriété de l'affichage .....	17
I.4.2 Réparer facilement Windows XP .....	20
I.4.3 Arrêt/Redémarrer/Mettre en veille.....	20
I.4.4 Gestion de la calculatrice .....	21
I.4.5 Gestion du clavier .....	22
I.4.6 Gestion de la Souris.....	22
I.5 Introduction au système d'exploitation Unix et Linux .....	24
I.5.1 Définition .....	24
I.5.2 Histoire de l'UNIX.....	25
I.5.3 Histoire du Système LINUX et sa mascotte.....	25
I.5.4 Avantage du système Linux .....	26
I.5.5 Systèmes Unix-Like.....	27
I.5.6 Principales distributions de Linux.....	28
I.5.7 Commande descriptives de Linux .....	28
I.5.8 Structure de Linux .....	31
I.5.9 Début de session.....	31
I.6 Liste de références bibliographiques .....	31

## I.1 Introduction

Windows est un système d'exploitation, c'est grâce à lui qu'on pourra faire comprendre à l'ordinateur ce qu'il doit exécuter. Il met à la disposition de l'utilisateur une interface graphique (bureau, icônes, boutons, fenêtres, menus, ...) qui permet de donner des ordres au micro-ordinateur de manière simple et conviviale.

Windows évolue en versions successives : 95, 98, NT, Me, 2000, XP. Pour une raison de compatibilité, un autre système d'exploitation plus ancien se cache encore derrière Windows, il s'agit du DOS. Il existe d'autres systèmes d'exploitation, comme par exemple LINUX.

Chaque programme fonctionnant sous Windows va « s'ouvrir » dans une fenêtre. On peut ainsi ouvrir plusieurs fenêtres les unes sur les autres à la fois (plusieurs programmes simultanément). Cela signifie que Windows est un système multitâche (imprimer un document pendant qu'on tape un texte, par exemple).

## I.2 Évolution des versions de Windows

Implanté à Seattle, la firme Microsoft a développé et mis sur le marché le système d'exploitation Windows. Même-si son activité commerciale commence avec le MS-DOS. Par souci chronologique, il convient de présenter les faits d'une manière linéaire suivant le tableau ci-dessous :

- **1990, Windows 3.1 et 3.11** : Les premières versions réellement adoptées par les utilisateurs. Ce ne sont qu'une interface, pas un OS. Le programme est du code 16 bits.
- **1993, Windows NT 3.1** : Est une version imaginé pour le monde professionnel (sécurité) et permettre à profit les instructions 32 bits. Reçoit le n° de version 3.1 pour remplacer Windows 3.1. Il réclamait malheureusement plus de mémoire et les applications 32 bits n'étaient pas disponibles (Echec commercial).
- **1995, Windows 95 / MS-DOS 7.0** : Ressemble plus à un OS. Il démarre sans passer par MS-DOS. C'est du code 32 bits mais pas entièrement. Windows 95 n'est toujours pas un système d'exploitation mais il contient déjà des fonctions propres aux prochains OS : Multitâche et Mémoire virtuelle. Le système de fichier reste celui du DOS.
- **1996, Windows NT 4.0** : Adopte une interface utilisateur analogue à celui de Windows 95. L'OS est écrit en C pour être portable (Alpha, Power PC) L'interface utilisateur est écrite en C ++.

➤ **1998, Windows 98 / MS-DOS 7.1.** : Cette version a été nécessaire à cause de la réticence des utilisateurs à migrer vers NT. La version 98 contient du code 16 bits + 32 bits mais apporte cependant quelques nouveautés :

- ✓ Le système de fichier FAT32
- ✓ Le plug and play.

➤ **2000, Windows Me Millenium Edition** : Les utilisateurs hésitent toujours à passer sur NT. Windows ME présente quelques améliorations au niveau des fonctions multimédia + la possibilité de restaurer le système dans un état précédent. Cette version a été fort décriée par les utilisateurs qui n'y ont vu qu'une mise à jour mineure de Windows 98. Elle était considérée comme « peu stable ».

**Windows 2000** : «2000» au lieu de Windows NT 5.0 pour que les utilisateurs y voient un successeur de Windows Me/9x et avoir un système d'exploitation unique. C'est un vrai système d'exploitation qui tourne en mode noyau.

- ✓ Multithread
- ✓ Capable de gérer plusieurs processeurs
- ✓ La mise en cluster est possible (groupe de machines qui travaillent comme une seule)
- ✓ Support du plug and play
- ✓ NTFS + Cryptage des fichiers
- ✓ Active directory pour gérer les comptes des utilisateurs
- ✓ Un code unique avec les langues dans un répertoire séparé
- ✓ Une base de données interne : le « Registre »

**2002, Windows XP** : Une nouvelle tentative pour faire oublier Windows 9x. Cette version parvient enfin à rassembler tous les utilisateurs. Fondée sur Windows 2000 se décline en plusieurs versions : Familiale, Professionnelle, Server et Advanced Server.

- ✓ Nouvelle interface
- ✓ Amélioration au niveau multimédia (pour le grand public)
- ✓ Prise en charge de lecteurs DVD
- ✓ Création de CD audio
- ✓ Pour les images affichage de miniatures pour les répertoires + Diaporama
- ✓ Lien plus étroit au WEB
- ✓ Pare feu
  
- ✓ Dossiers WEB

- ✓ Activation des produits par Internet
- ✓ Assistance à distance et d'autres « services WEB » ;

➤ **2007 : Windows Vista** : Parmi les nombreuses nouveautés de cette version les utilisateurs aurons remarqué :

- ✓ Une nouvelle interface, nommée Aero qui tire parti des cartes graphique puissantes, pour afficher des effets 3D et de transparence
- ✓ La récupération automatique à la suite d'un problème au démarrage ou d'un service en échec.
- ✓ Windows Update apparait comme faisant partie du panneau de configuration et non plus comme un logiciel en ligne.
- ✓ Des fonctions de recherche plus développée grâce à l'indexation des fichiers "Recherche pendant la frappe"

Cette version a déçu les utilisateurs car trop gourmande en ressources elle s'est avérée lente et instable. Certains utilisateurs sont repassés à XP et bon nombre d'entreprises peu convaincues par les nouveautés de Vista n'ont pas jugé nécessaire de changer de version.

➤ **2009 : Windows 7** : Successeur réussi de Vista, dès son installation Windows 7 semble plus rapide et est donc perçu un produit plus performant. Windows 7 parvient à tourner sur des machines plus légères ce qui est souvent le cas pour les PC portables qui remplacent le plus en plus souvent les PC de bureau. Cette version a aussi été développée pour s'adapter plus facilement au WiFi. Voici quelques autres modifications retenues quant à l'ergonomie de cette version :

- ✓ La barre des tâches a été modifiée : les boutons de tâches sont remplacés par des icônes sans texte. Les icônes qui correspondent à une même application ouverte dans plusieurs fenêtres se superposent. La liste des documents ouverts surgit comme un menu déroulant lorsqu'on survole ces icônes avec la souris.
- ✓ Les bibliothèques permettent de regrouper dans un même affichage des fichiers qui sont stockés en divers emplacements.
- ✓ Snap offre une nouvelle manière pour redimensionner les fenêtres ou les disposer plus facilement sur le bureau.
- ✓ Le Groupe résidentiel est une nouveauté pour interconnecter plus simplement les PC Windows 7 d'un réseau domestique afin de faciliter le partage des imprimantes et des bibliothèques

### I.3 Système de gestion de Windows

L'écran de Windows XP peut-être différent d'un poste à l'autre tout dépend :

- ✓ des paramètres de Windows XP (fonds d'écrans, couleurs, mise en veille),
- ✓ des versions de Windows (95, 98, 2000, Me, XP fam, XP pro ...),
- ✓ des raccourcis et des programmes installés par l'utilisateur. Cependant les principes présentés dans ce chapitre restent les mêmes dans tous les cas.

Les objets de l'interface graphique WINDOWS XP sont :

- ✓ des icônes
- ✓ des fenêtres
- ✓ des boites de dialogue

#### I.3.1 Bureau ou Desktop (dessus de bureau)

##### A. Démarrer Windows XP / Le Bureau

Pour démarrer Windows, nous devons suivre les étapes suivantes :

1. Mettre l'ordinateur sous tension (utiliser le Grand bouton sur la face avant de l'unité) : Le système d'exploitation Windows XP se charge automatiquement.
2. Un écran apparaît : Comme il vous est demandé, appuyez sur les touches : **Ctrl + Alt et Suppr.**

- Il vous faut à présent vous connecter au réseau ; pour cela on doit saisir l'identifiant et le mot de passe puis taper simplement sur la touche **[Entrée]** du clavier (ne taper pas sur Annuler, ou la touche [ Echap ] ).
- Le bureau apparaît sur l'écran (voir figure I.1): Le bureau représente votre environnement de travail. Il supporte des icônes (objets) et la barre des tâches.

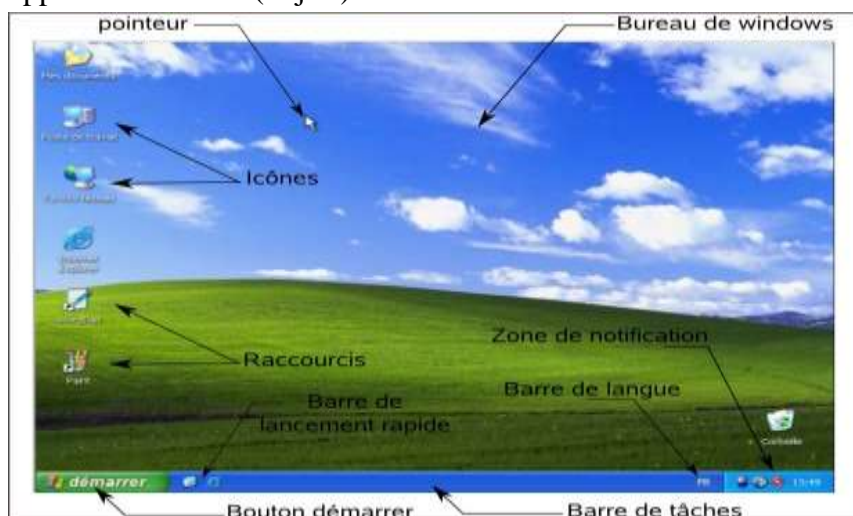


Figure I.1 : Bureau de l'environnement Windows.

Lorsque votre ordinateur est mis en route, Windows démarre automatiquement. Plusieurs éléments apparaissent alors sur votre « bureau » qui représente notre environnement de travail. Il supporte des icônes (objets) et la barre des tâches.

## B. Les icônes

Les icônes sont des boutons sous forme des petites images permettant d'accéder à des fenêtres (programmes). On distinguera deux types d'icônes :

- ✓ Les icônes du matériel lui-même (Poste de travail, Explorateur Internet (ou tout autre application), corbeille, etc.).
- ✓ Les icônes des programmes (Word, Excel, Access, etc.).

Enfin il peut y avoir des dossiers et même des documents qui ont été déposés sur le bureau. Il faut cliquer deux fois sur une icône pour démarrer un programme.

Exemple : ci-dessous deux icônes qu'on peut retrouver probablement sur le bureau.



Le « **poste de travail** » permet de voir les différents composants de votre ordinateur et de gérer les fichiers.



La « **Corbeille** » est un lieu de stockage temporaire pour les fichiers supprimés, elle permet de récupérer des fichiers supprimés par erreur.

Les autres icônes sont des raccourcis qui permettent de lancer (démarrer) les programmes installés sur notre PC. On peut rajouter autant de raccourcis que l'on veut sur le bureau. L'élimination des icônes ne signifie pas que les programmes correspondants disparaissent du disque dur. Le tableau ci-dessous indique quelques manipulations des icônes.

Objectif	Méthode
Sélectionner une icône	Un simple <b>Clic Gauche</b> sur l'icône.
Déplacer une icône	Sélectionner l'icône, mais <b>Maintenir</b> le bouton gauche de la souris et <b>glisser</b> vers le nouvel emplacement.
Supprimer une icône	Sélectionner l'icône, taper sur la touche [ <b>Suppr</b> ] de votre clavier. Confirmer par la touche O ou en cliquant sur OK.

## C. Les raccourcis

Comme le montre la figure I.2, les raccourcis sont des liens vers des fichiers ou des applications que l'on place sur le bureau. Ils se présentent comme des icônes avec une petite flèche courbe dans le coin inférieur gauche. La suppression d'un raccourci fait simplement disparaître le lien et non pas le dossier ou le programme qu'il désigne. Certains raccourcis peuvent être placés sur la barre des tâches dans la zone dite de "lancement rapide" (XP) ou



"programmes épinglés" (Windows 7). Ils demeurent de cette manière toujours visibles à l'écran même quand les fenêtres d'applications recouvrent la surface du bureau. Il est aussi possible de faire glisser ces raccourcis vers le bouton démarrer de sorte à les "épingler" dans la partie supérieure du menu démarrer.



Figure I.2 : Raccourcis sous Windows.

Pour ajouter un raccourci vers un programme dans la barre, ouvrez le programme. Ensuite, faites un clic droit sur son icône dans la barre des tâches et sélectionnez l'option « Épingler ce programme à la barre des tâches ».

#### D. la barre de tâches et le menu démarrer

Au bas du bureau il y a la « barre de tâches » qui contient généralement cinq parties comme l'indique la figure I.3.

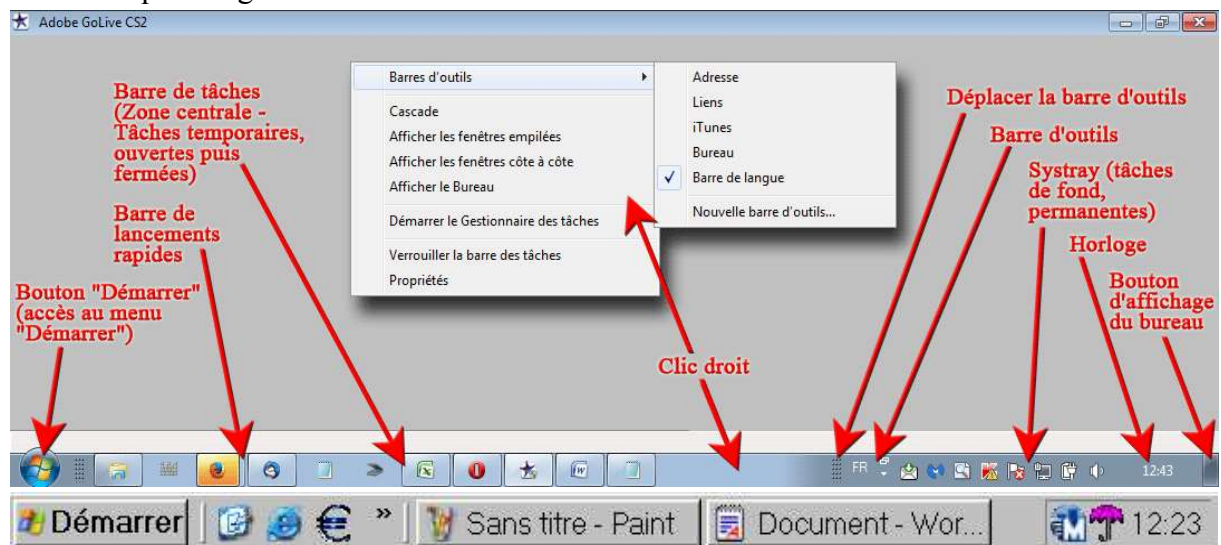


Figure I.3 : Barre des tâches et Menu Démarrer sous Windows xp.

- Dans sa partie gauche, on peut cliquer sur le bouton « **Démarrer** » pour ouvrir le menu qui nous permet de lancer des programmes, de consulter l'aide de Windows, de gérer et rechercher des documents et enfin, d'arrêter Windows.
- Cliquez sur « **Tous les programmes** » du menu « Démarrer » pour afficher la liste de tous les programmes installés sur votre ordinateur. Dans la zone inférieure, un champ de

recherche permet de trouver rapidement un fichier, dossier ou programme sur votre ordinateur.

- En bas à droite, cliquez sur le bouton "**Arrêter**" pour éteindre votre ordinateur ou cliquez sur la flèche à sa droite pour Changer d'utilisateur, Mettre en veille ou Redémarrer.
- A côté du bouton « **Démarrer** » apparaîtront de nouveaux boutons (un bouton par application ouverte) pour faciliter le passage d'une fenêtre à une autre en cliquant tout simplement sur le bouton de l'application désirée pour que sa fenêtre s'affiche au premier plan.
- **La zone de notification** se trouve à droite de la barre des tâches. Elle inclut une horloge et un groupe d'icônes. Elle vous permet d'accéder à des réglages de configuration un peu plus poussés : son, réseau, antivirus, périphériques ... L'ensemble d'icônes affiché dépend des programmes et services installés, et de la façon dont votre ordinateur a été configuré par le fabricant. Afin de gagner de la place, toutes les icônes ne sont pas affichées. Cliquez sur la petite flèche pour les voir. Si vous avez ajouté un nouveau périphérique matériel sur votre ordinateur (clé USB, par exemple), une icône peut apparaître dans cette fenêtre pour vous notifier l'événement. Avant d'enlever votre périphérique USB de votre ordinateur, cliquez sur cette icône et sélectionnez l'option « Éjecter + Nom de votre périphérique ». Un message vous informera ensuite que vous pouvez enlever votre clé. Sans cette manipulation, vous risquez d'endommager votre périphérique.

### I.3.2 Le poste de travail

Le poste de travail est l'accès principal à toutes les unités de stockage tel que : le lecteur de disquette, disques durs, graveur de CD, disques amovibles, cartes mémoire etc...

Windows occupe lui-même une place qu'il se réserve sur le disque dur. Par défaut, il se trouve toujours dans le disque dur local C, dans le répertoire « Windows ».

Le poste de travail ressemble bien un meuble de bureau qui s'organise en tiroirs qui contiennent des répertoires, qui eux-mêmes contiennent des dossiers etc.... Le Poste de travail donc contient à son tour, un ou plusieurs disques durs qui contiennent des répertoires, qui contiennent des dossiers qui peuvent à leur tour contenir des sous-dossiers. Etc...

Ce mode de classement en arbre qu'on appelle aussi arborescence Windows permet de classer et retrouver les documents, du moment qu'on connaît leur chemin (leur emplacement dans l'arborescence).

Exemple :

C : Windows\Fonts\Arial désigne le fichier dénommé « Arial » qui se trouve dans le dossier « Fonts » du répertoire « Windows » qui se trouve dans le disque dur « C ».

Pour démarrer le poste de travail, on clique sur le bouton «démarrer» dans la barre des tâches, puis sur Poste de travail. La fenêtre poste de travail s'affiche.

### **A. Fichier Informatique**

Un fichier informatique est une unité informationnelle physiquement stockée sur un support de mémoire de masse permanent (disque dur par exemple). Un fichier a généralement comme attributs : un nom ; un chemin d'accès ; une taille mesurée en octets ; une date de création et une de dernière modification. Plusieurs systèmes de fichiers multi-utilisateur rattachent aussi à chaque fichier un propriétaire et des droits d'accès. Enfin, chaque fichier a un contenu, soit une suite ordonnée d'octets, qui peut représenter n'importe quelle donnée binaire déterministe : un programme informatique, un document, un texte, etc.

### **B. Quelques dossiers trouvés sur le disque C**

Windows qui contient le système d'exploitation

- ✓ Program Files (fichiers-programmes) qui contient les « applications » (comme Word, Excel...
- ✓ Mes documents qui nous est réservé en tant qu'utilisateur, afin d'y stocker nos propres données. Dans certaines versions, « Mes Documents » se trouve dans « Documents and Settings »).

### **C. Quelques types de fichiers à connaître**

A l'intérieur des dossiers, les fichiers contiennent les données à proprement parler. Il existe plusieurs types de données. Chaque type de donnée se distingue par une extension : un suffixe en principe composé de 3 lettres, par exemple « .doc »

- ✓ .doc document au format Word (texte)
- ✓ .exe application (programme exécuté par la machine, par exemple Winamp.exe)
- ✓ .zip ensemble de données compressées, par exemple des fichiers d'installation
- ✓ .mp3 fichier musical compressé
- ✓ jpg fichier image compressé

### **D. Association de fichiers**

Chaque application est capable de lire certains types de fichiers :

- ✓ l'application Winamp pourra lire les fichiers musicaux .mp3

- ✓ l'application Word pourra lire les fichiers de texte au format .doc
- ✓ l'application Winzip pourra décompresser les fichiers compressés de type .zip

### I.3.3 La fenêtre d'une application (d'un programme) sous Windows

Les programmes, les documents, les messages d'erreur, les messages d'information dans Windows s'ouvrent dans des fenêtres. Ces fenêtres présentent toute une série d'éléments communs. Nous concentrons avec la figure I.4 pour les découvrir.

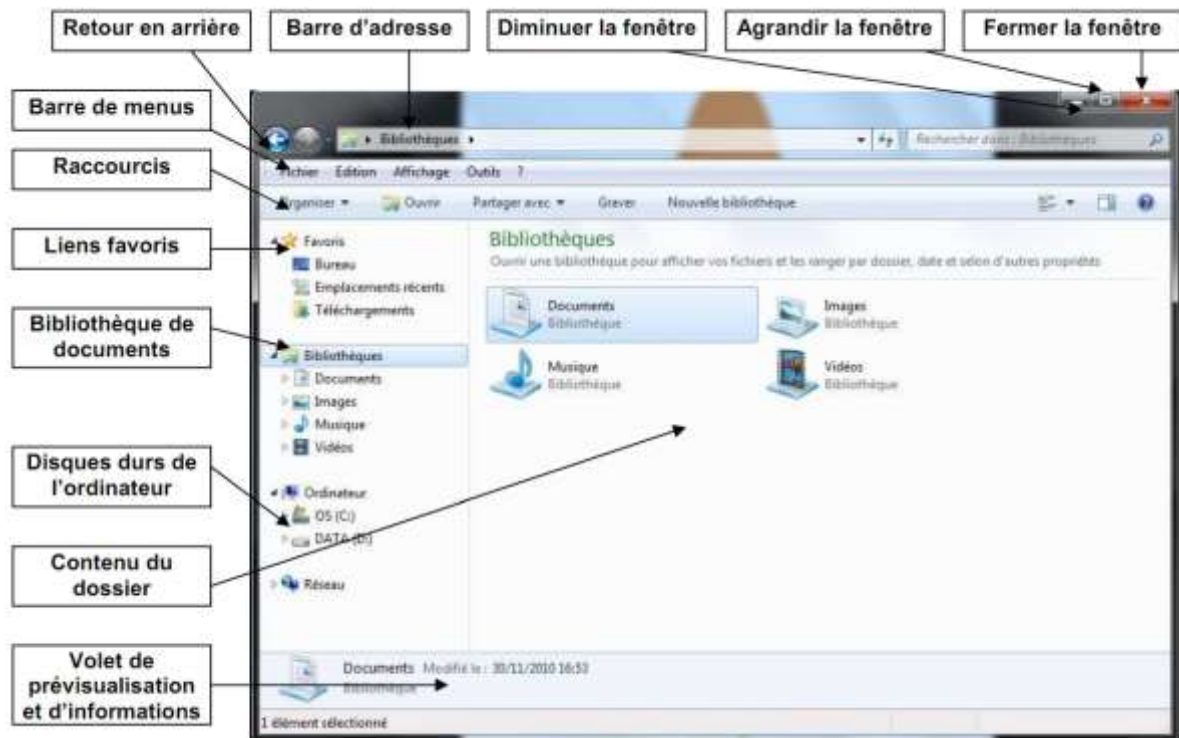


Figure I.4 : Fenêtre sous l'environnement Windows.

#### A. Barre des menus

La barre des menus, donne accès aux différents menus proposés par le programme. Dans cette barre on peut trouver plusieurs menus déroulants : il suffit de cliquer sur un menu pour qu'il s'ouvre (qu'il se déroule...) et de cliquer ensuite sur l'option désirée. C'est par le menu « Fichier » que l'on peut : ouvrir, enregistrer ou imprimer un travail. Le petit triangle noir permet d'afficher l'arborescence de votre poste de travail afin de sauvegarder vos données au bon emplacement.

#### B. Barre de titre

La barre de titre, indique le nom du document et du programme dans lequel vous vous trouvez.

### C. Bouton Réduire

En cliquant sur le bouton « Réduire », nous réduisons la fenêtre. Elle ne sera plus visible que dans la barre des tâches.

### D. Bouton Agrandir

Ce bouton permet d'agrandir la fenêtre. Après l'avoir agrandi, l'icône change et permet de revenir à la taille initiale.

### E. Bouton Fermer

En cliquant sur cette croix, vous fermez la fenêtre et le programme.

### F. Barre de défilement

La barre de défilement permet de faire défiler le contenu de la fenêtre afin d'afficher les informations qui ne sont pas actuellement visibles à l'écran (Figure I.5).



Figure I.5 : Barre de défilement.

**F.1. Flèche de défilement :** cliquez brièvement de façon répétée sur la flèche de défilement vers le haut ou vers le bas pour faire défiler le contenu de la fenêtre dans la direction souhaitée. Maintenez le bouton de la souris enfoncé pour faire défiler la fenêtre plus rapidement.

**F.2. Ascenseur :** Pour faire défiler le contenu d'une page, on utilise les ascenseurs. Ceux-ci se déplacent dans la barre de défilement. Nous glissons l'ascenseur vers le haut, le bas, la gauche ou la droite pour faire défiler la fenêtre dans la direction souhaitée.

**F.3. Barre de défilement verticale :** cliquez sur une zone vide d'une barre de défilement, au-dessus ou en dessous de l'ascenseur pour faire défiler une page vers le haut ou vers le bas.

**F.4. Barre de défilement horizontale :** cliquez sur une zone vide d'une barre de défilement, à gauche ou à droite de l'ascenseur pour faire défiler une page vers la gauche ou la droite.

### G. Bordures et coins

Ce type de bordures est utilisé pour modifier la taille de la fenêtre.

#### ✓ Déplacement d'une fenêtre

Pour déplacer une fenêtre, utilisez la barre de titre (c'est celle qui contient les boutons).

✓ Cliquez sur cette barre et maintenez le bouton gauche enfoncé.

✓ Ensuite, déplacez votre souris et vous verrez la fenêtre bouger.

## H. Affichage en plein écran

Pour qu'une fenêtre s'affiche en plein écran, déplacez la fenêtre vers le haut de l'écran.

### ✓ Redimensionnement d'une fenêtre

Pour modifier la taille d'une fenêtre manuellement, il faut :

1. Placer le pointeur de la souris le bord d'une fenêtre à partir duquel on souhaite la redimensionner. Le pointeur se modifie alors pour prendre l'apparence d'une flèche à deux têtes :



2. Appuyer alors sur le bouton gauche de la souris, le garder enfoncé, faire glisser la souris dans la direction de son choix.

Lors de cette opération, une bordure s'affiche à l'écran pour montrer la taille que prendra la fenêtre si on relâche le bouton de la souris. On peut effectuer aussi cette opération à partir des coins de la fenêtre.

### ✓ Pour passer d'une fenêtre à l'autre

Pour passer d'une fenêtre à une autre, on fait un **simple clic** n'importe où sur la fenêtre que l'on désire activer. On peut aussi utiliser l'**indicateur de la fenêtre** qui se trouve dans la barre des tâches. Un simple clic dessus suffit.

### ✓ Afficher toutes les fenêtres ouvertes à l'aide des options mosaïques

- Toutes les fenêtres doivent être ouvertes.
- Les fenêtres fermées ou réduites ne seront pas affectées par cette méthode.
- Les fenêtres peuvent se superposer les unes sur les autres.
- Pour accéder aux options mosaïques :
  - ✓ faire un **Clic Droit** sur une zone vide de la barre des tâches
  - ✓ Un menu contextuel s'ouvre, sélectionner avec un **Clic Gauche** sur **Mosaïque verticale**.

- ✓ De même, afin de restaurer l'état initial des fenêtres, il faut faire un **Clic Droit** sur une zone vide de la barre des tâches, et sélectionner avec un **Clic Gauche** Annuler Mosaïque.

### I.3.4 Explorateur Windows

L'Explorateur est un programme qui permet de gérer le disque dur (changer le nom, copier, supprimer ou déplacer des fichiers). On lance l'Explorateur en cliquant deux fois sur son icône située sur le bureau ou dans le menu « Démarrer ». On l'atteint également par clic droit de la souris, sur n'importe quel disque ou dossier, en choisissant l'option « Explorer ». La fenêtre de l'explorateur Windows est représentée par la figure I.6.

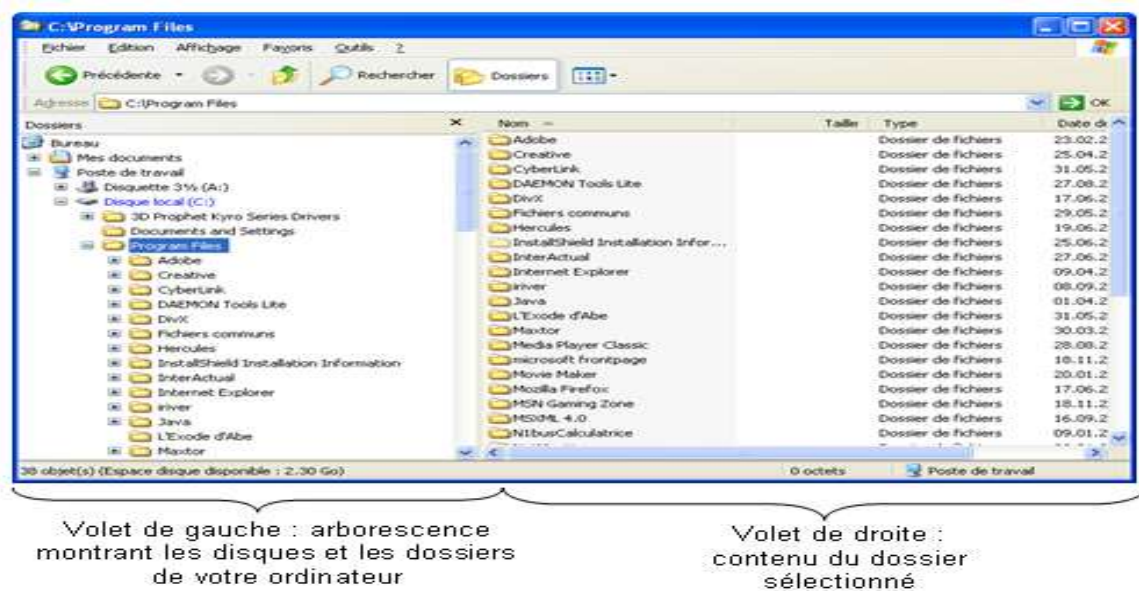
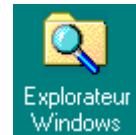


Figure I.6 : Fenêtre explorateur Windows.

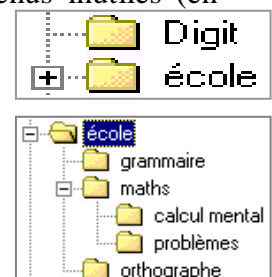
La fenêtre qui apparaît se divise en deux grandes zones :

#### A. Partie Gauche

La partie gauche représente « l'arborescence » de la machine. On peut créer de nouveaux dossiers ou sous-dossiers autant qu'on le désire (en sélectionnant « Nouveau » du menu « Fichier » et ensuite, « Dossier ») et supprimer ceux qui sont devenus inutiles (en sélectionnant « Supprimer » du menu « Fichier »).

Le petit « + » à côté d'un dossier signifie qu'il contient un ou plusieurs sous-dossiers. Cliquer sur ce « + » permet de dérouler les sous-dossiers.

Le classement peut ainsi se faire de manière très cohérente (par exemple, un dossier appelé « école » contiendra des sous-dossiers appelés « grammaire », « maths », « orthographe », ... et le sous-dossier « maths »



peut lui aussi contenir d'autres sous-dossiers appelés « calcul mental », « problèmes », ... et ainsi de suite).

L'icône d'un dossier sélectionné change de forme, il « s'ouvre » et son nom change de couleur.

## B. Partie droite

La partie de droite affiche le contenu du dossier sélectionné : Dans cette partie, on peut : activer (lancer) un programme ou sélectionner...

Grace à l'explorateur Windows, on peut créer dans l'arborescence de fichier, des nouveaux répertoires pouvant contenir de nouveaux fichiers (clic droit puis choisissez « nouveau dossier »)

On peut aussi « naviguer » dans cette arborescence :

- ✓ Ouvrir un fichier
- ✓ Supprimer un fichier (clic droit ou « suppr » ou « shift-suppr »)
- ✓ Déplacer des fichiers/répertoires par simple « glisser »
- ✓ Copier/couper/coller des fichiers/répertoires (ctrl-c -x -v)
- ✓ Afficher les propriétés du fichier/répertoire (clic droit)
- ✓ Rechercher un (ou des) documents
- ✓ Découvrez ce que comportent les menus...

### I.3.6 Rechercher des fichiers sous Windows

Des fois on stock nos fichiers rapidement sans faire attention à l'emplacement de rangement. Windows possède un outil de recherche très pratique comme le montre la boîte de dialogue de la figure I.8. Il est accessible à partir du menu « Démarrer ».

Cliquer sur « Rechercher » puis sur « *Fichiers ou dossiers* » offre des critères de recherche :

- ✓ Par nom (onglet « Nom et emplacement ») ou par morceau de nom
- ✓ Par fichiers créés sur une période donnée (Onglet « Date »).
- ✓ basée sur le contenu d'un fichier (onglet « Avancée »)



L'opération de recherche sera plus facile et rapide, si vous avez donné un nom cohérent à votre fichier.

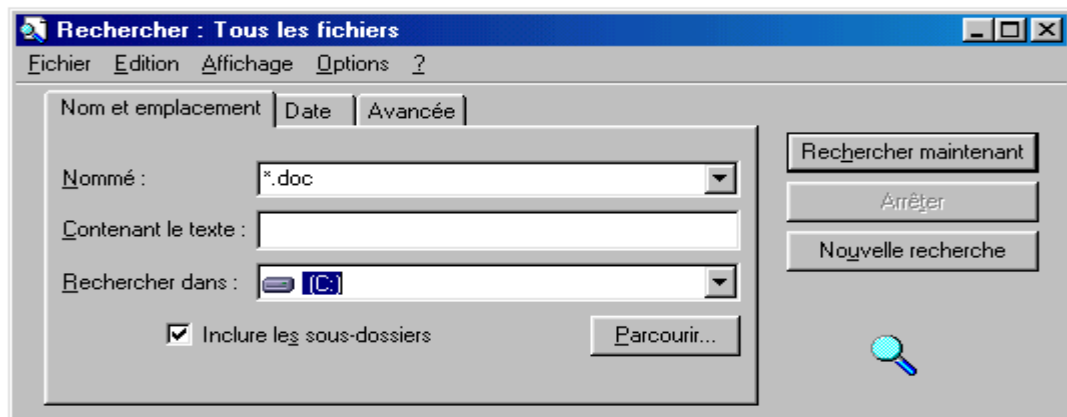


Figure I.8 : Fenêtre de recherche sur Windows.

### I.3.7 Panneau de configuration

Le panneau de configuration est un élément important de Windows puisque c'est par son intermédiaire qu'on peut modifier ou personnaliser certaines options de configuration de notre système. Pour accéder le panneau de configuration, il y a deux manières :

- soit en passant par le « Poste de travail » et en cliquant sur l'icône « Panneau de configuration ».
- soit en suivant le cheminement bouton « Démarrer » – « Paramètres » – « Panneau de configuration ».



La figure I.8, représente le contenu des principales options du Panneau de configuration :

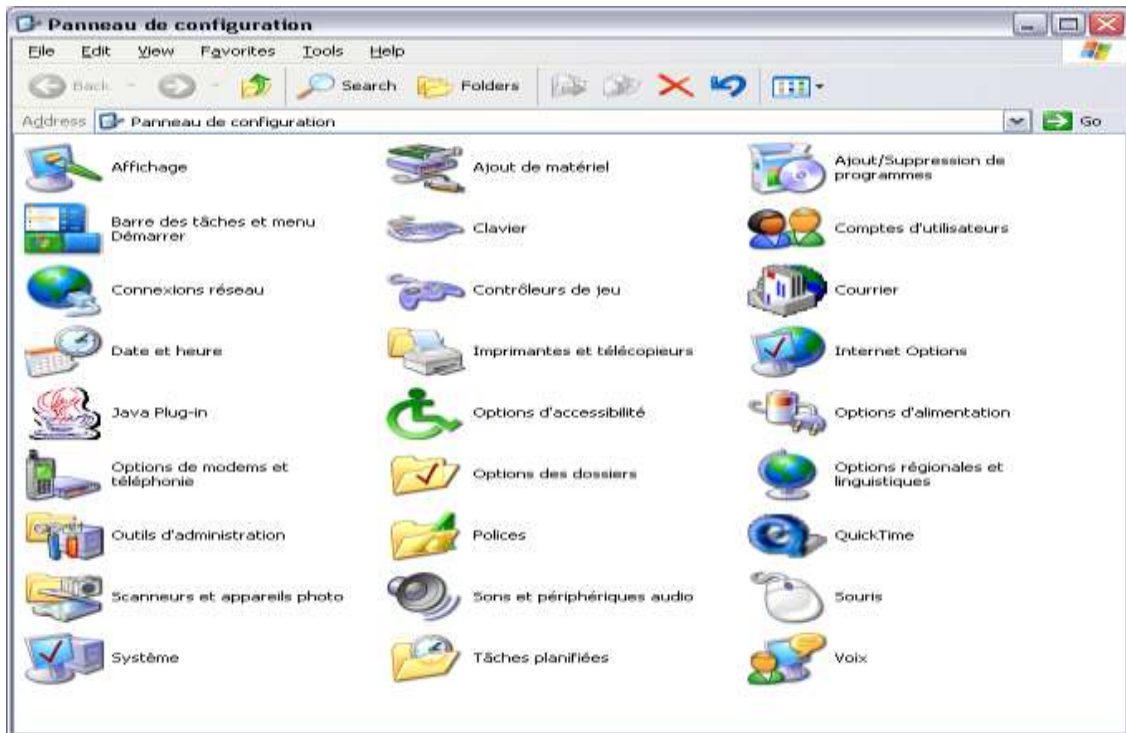























Figure I.8 : Fenêtre panneau de configuration sous Windows.

Les éléments sont regroupés par catégories : Système et Sécurité, Réseau et Internet, Programmes, etc. Cliquez sur le nom d'un thème pour faire apparaître dans une nouvelle fenêtre tous les réglages qu'il propose. Par ailleurs, dans la fenêtre principale, chaque catégorie propose déjà un, deux ou trois liens rendant directement accessibles les principaux réglages. Le panneau de configuration permet d'accéder à environ tous les éléments de l'ordinateur. Le tableau ci-dessous collecte le rôle de chaque icône du panneau de configuration.

 Comptes d'utilisateurs	Ce bouton permet de connecter avec un nom d'utilisateur et un mot de passe. Le changement du mot de passe et le nom d'utilisateur à partir de ce bouton est possible.
 Affichage	Ce bouton permet de faire des modifications à l'affichage du bureau de travail
 Ajout de matériel	Permet de faire l'installation de nouveaux périphériques tels qu'imprimante, graveur, etc.
 Ajout/Suppression de programmes	Permet de faire des ajouts ou des suppressions de programmes
 Clavier	Permet de modifier la langue du clavier ainsi que la vitesse du curseur
 Connexions réseau	Permet de faire la configuration pour les connexions réseaux
 Contrôleurs de jeu	Permet d'installer et faire la configuration des manettes de jeux

 Courrier	Permet de configurer les options de comptes courriers pour Outlook
 Date et heure	Permet d'ajuster l'heure et la date de l'ordinateur
 Imprimantes et télécopieurs	Permet de faire la configuration des imprimantes et télécopieurs
 Internet Options	Permet la configuration de l'Internet.
 Barre des tâches et menu Démarrer	Permet de personnaliser la barre de tâche et le menu démarrer.
 Options de modems et téléphonie	Permet modifier les options du modem
 Options d'accessibilité	Permet de changer l'interface de Windows pour les personnes ayant des déficiences visuelles
 Options d'alimentation	Permet de changer les paramètres d'alimentation de votre ordinateur.
 Options des dossiers	Permet de personnaliser plusieurs options sur les dossiers : ouverture, enregistrement, création, etc.
 Options régionales et linguistiques	Permet de modifier les options régionales et linguistiques de Windows.
 Outils d'administration	Permet de gérer l'ordinateur
 Polices	Permet de voir les polices de caractères déjà installés sur votre ordinateur. C'est à cet endroit qu'on peut supprimer ou en ajouter.
 Scanneurs et appareils photo	Permet de faire la configuration des scanners ou des appareils photo
 Souris	Permet de faire la configuration de la souris, tel que les boutons gauches/droites.

## I.4 Opérations de gestion sous Windows

### I.4.1 Propriété de l'affichage

Pour pouvoir accéder au propriété de l'affichage, on doit ressortir le menu contextuel en faisant un bouton de droite directement sur le bureau et choisir Propriété dans le menu défilant. Les propriétés de l'affichage permettent à l'utilisateur de :

- Changer les thèmes de Windows (Onglet Thème)
- Modifier l'image d'arrière-plan du bureau (Onglet Bureau)
- Modifier l'écran de veille (Onglet Écran de veille)
- Changer l'apparence de Windows (Onglet Apparence)
- Changer la résolution l'affichage de Windows (Onglet Paramètres)

### A. Onglet Thème

Il est possible de changer l'apparence de Windows en choisissant un thème prédéfini par Windows. On peut changer les thèmes autant de fois que vous le souhaitez et vous pouvez également télécharger directement sur Internet d'autres thèmes fournis par Microsoft.



Figure I.9 : Onglet 'Thème' de la fenêtre « propriétés d'affichage ».

### B. Onglet Bureau

Il est possible de changer l'image de bureau. Il faut choisir l'image qui convient le mieux. Si aucune image ne nous intéresse, nous pouvons choisir nous-même une image que nous avons déjà dans notre ordinateur. Pour ce faire, nous devons appuyer sur Parcourir et aller chercher l'image à l'endroit où il est situé dans l'ordinateur. L'option position vous permet de positionner votre image. Trois choix sont possibles : Étirer, Centrer et Mosaïque.



Figure I.10 : Onglet 'Bureau' de la fenêtre 'propriétés d'affichage'.

### C. Onglet Écran de veille

Nous pouvons changer l'écran de veille de l'ordinateur par les écrans de veille déjà fournis par Microsoft. Il est important de noter que nous pouvons aussi définir le temps pendant lequel Windows commence à afficher l'écran de veille. Par exemple, si vous avez spécifié 10 minutes, alors après 10 minutes d'inactivité, Windows affiche l'écran de veille afin d'économiser l'énergie de votre écran.



Figure I.11: Onglet 'Ecran de veille' de la fenêtre 'propriétés d'affichage'.

### D. Onglet Apparence

L'onglet apparence permet de changer complètement l'interface de travail de Windows. Trois modèles sont disponibles et on peut choisir celui qui nous convient le mieux. Windows offre la possibilité de personnaliser notre modèle en cliquant sur Avancé. On peut alors changer les polices, la taille des icônes, la couleur ...

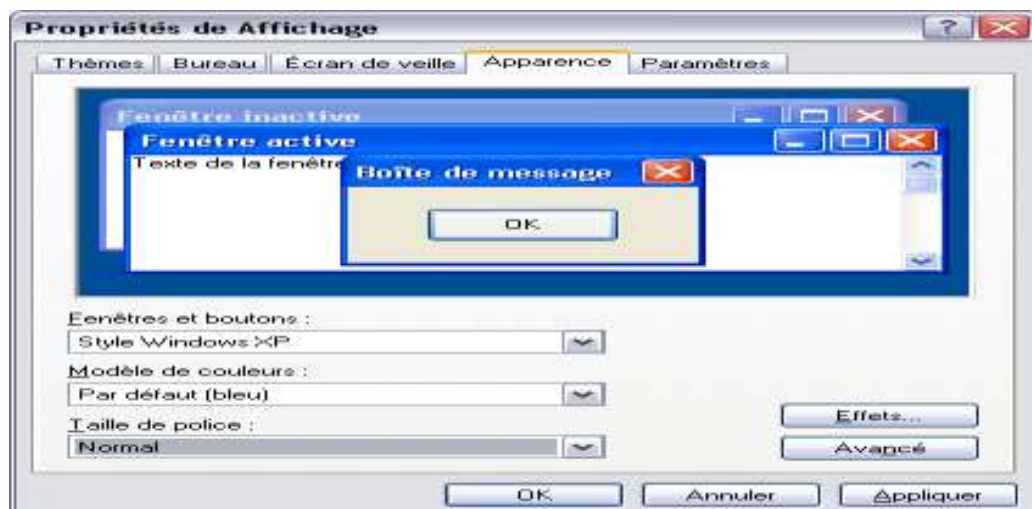


Figure I.12 : Onglet 'Apparence' de la fenêtre 'propriétés d'affichage'.

### E. Onglet Paramètres

L'onglet paramètres est très important car elle permet d'accéder aux options de la carte graphique. Ainsi, on peut changer la résolution de la surface de travail et surtout changer la fréquence de balayage de l'écran.

Conseil pour le paramétrage :

- **Pour un écran de 15 pouces :**
  - ✓ Résolution 800 X 600
  - ✓ Couleur : 32 bits
- **Pour un écran de 17 pouces :**
  - ✓ Résolution 1024 X 768
  - ✓ Couleur : 32 bits
- **Pour un écran de 19 pouces :**
  - ✓ Résolution 1600 X 1200
  - ✓ Couleur : 32 bits



Pour la fréquence de balayage, il vaut mieux de choisir 75 hertz parce que la plupart des cartes graphiques supporte cette fréquence. Pour changer la fréquence de balayage, vous devez aller dans Avancé et choisir l'onglet Écran.

### I.4.2 Réparer facilement Windows XP

Cette méthode est utile lorsque Windows devient trop instable qui demande la restauration du système.

- Cliquez sur Démarrer > Tous les programmes > Accessoires > Outils système > Restauration du système.
- Sélectionnez un point de restauration sur le calendrier où vous pensez que tout allait encore bien à cette époque.
- Cliquez sur suivant puis sur fermer.

### I.4.3 Arrêt/Redémarrer/Mettre en veille

#### A. Quitter Windows XP

Quand on termine notre travail sur le PC, nous allons quitter Windows XP et arrêter l'ordinateur. Avant toute chose on doit sauvegarder notre travail (**Fichier, Enregistrer Sous**, etc.), puis on ferme toutes les fenêtres (avec le bouton [X] en haut à droite de chaque fenêtre) et enfin on passe par deux étapes :

- ✓ un Clic Gauche sur le bouton **Démarrer** : (en bas à gauche de l'écran)

- ✓ Puis un clic sur **Arrêter l'ordinateur**.

l'écran de la figure I.13. Apparaît, on fait un Clic Gauche sur **Arrêter**.



Figure I.13 : Fenêtre Arrêt/Redémarrer/Mettre en veille de l'ordinateur.

## B. Redémarrer

Suite souvent à un blocage ou installation d'un nouveau logiciel, le redémarrage permet de remettre les choses en place ou de prendre en compte le nouveau logiciel installé.

## C. Mettre en veille

Utilisée en cas d'une longue pause. Il faut enregistrer les fichiers ouverts sur le bureau (pas obligé de les fermer) sinon en cas de coupure de l'électricité entre-temps on perdra tout le travail non enregistré.

- Les micro-ordinateurs récents sont dotés d'un arrêt automatique. Quitter Windows XP vous mettra hors tension, il est donc inutile d'appuyer sur le bouton **Power**. De plus l'écran peut rester allumé, au bout de quelques minutes il se mettra tout seul en veille.
- Il existe un raccourci clavier pour arrêter votre poste : **Ctrl + Alt + Suppr**. Etc.

## I.4.4 Gestion de la calculatrice

- Pour démarrer la calculatrice:
  - ✓ cliquer sur **Démarrer, (Tous les) Programmes, Accessoires, Calculatrice**.
- Pour calculer, nous avons deux méthodes :
  - ✓ soit cliquer sur les **nombres** et les **opérateurs**,
  - ✓ soit les saisir avec le pavé numérique.
- Pour conserver le résultat :
  - ✓ cliquer sur **Edition** dans la barre des menus, puis sur **Copier**. Pour récupérer ce résultat dans une autre application résultat cliquer sur **Edition**, puis sur **Coller**.

### I.4.5 Gestion du clavier

- Quand on combine deux touches (par exemple [ Ctrl ] + [ i ] ), on appuie sur la première ici [ Ctrl ], on garde le doigt appuyé, puis on frappe après la deuxième ici [ i ].
- Les caractères spéciaux comme « @ », « # », « ] », « { », etc. s'obtiennent en combinant la touche [Alt Gr] et la touche qui les présentent en bas à droite. Par exemple pour « @ » on tapera [Alt Gr] + [0] (le zéro de la deuxième rangée de touche.)
- Sur le pavé numérique, on travaillera toujours avec la touche **VERR NUM** activée. Si la touche n'est pas verrouillée, les touches ont les mêmes fonctions que celles de déplacement du curseur.
- Travailler en minuscule c'est mieux, on (touche [CAPS LOCK] blocage du clavier en majuscule désactivé.
- Pour gérer les majuscules :
  - ✓ Soit le texte ne comprend qu'une lettre en majuscule, on va alors appuyer simultanément sur la touche **Shift** (Majuscules) et la touche de la lettre.
  - ✓ Soit il y a plusieurs lettres en majuscules, on passera en majuscule avec la **touche [ CAPS LOCK ]** « blocage du clavier en majuscule » on saisira le texte ; enfin on reviendra en minuscule en tapant sur la touche **Shift** (Majuscules).

Le tableau ci-dessous, résume quelques raccourcis possibles par clavier

[F1]	Aide Windows	[Ctrl] +[C]	Copier
[F2]	Renommer un objet sélectionné	[Ctrl] +[X]	Couper
[F3]	Fenêtre de recherche	[Ctrl] +[V]	Coller
[Alt]+[F4]	Quitter un programme ou Windows	[Del]	Détruire l'objet sélectionné
[Ctrl]+[F4]	Fermer une fenêtre	[Maj] +[Del]	Détruire sans passer par la corbeille
[F5]	Rafraîchir le contenu d'une fenêtre	[Ctrl] +[Tab]	Passer d'un onglet à l'autre
[Maj] + [F10]	Menu contextuel	[Maj] +Fermer	Fermer un dossier et dossiers parents
[Alt] +[Esc]	Passer de fenêtres en fenêtres	[Alt] +[Entrée]	Propriétés de l'objet sélectionné
[Ctrl] +[Esc]	Ouvrir le menu « Démarrer »	[Alt] +[Tab]	Passer d'une application à l'autre

### I.4.6 Gestion de la Souris

#### A. Sélection + Action

La souris est le périphérique le plus utilisée pour la sélection par un simple clic gauche. Mais aussi pour les déplacements, les actions, etc. En effet, on sélectionne d'abord quelque chose, puis on demande après au programme d'agir sur cette chose. Un Double clic gauche permet la Sélection + Action.



### A.1 Utilisation de la souris.

- En utilisant la souris on évitera une position trop haute ou trop basse du poignet.
- Les doigts doivent être relaxés.
- L'index doit reposer en permanence sur le bouton de gauche, prêt à appuyer dessus.
- La souris se doit d'être en contact permanent avec le tapis.

### A.2 Les opérations élémentaires avec la souris

- Déplacer la souris sur le bureau : C'est l'opération qui consiste à bouger la souris, un indicateur (le pointeur) se déplace sur le plan de travail (le bureau).
- Clic sur le bouton gauche de la souris (Clic G): c'est l'opération qui permet de sélectionner un élément (« noircir »).
- Double Clic gauche : double clic de suite (sans temps mort) sur le bouton gauche. C'est l'opération qui permet de sélectionner et d'activer un élément.
- Cliquer et glisser avec le bouton gauche de la souris enfoncé : Cette opération permet de déplacer des objets; elle consiste à appuyer sur le bouton gauche de la souris puis à déplacer celle-ci en maintenant la pression sur le bouton. L'action s'arrête au moment où l'on relâche la pression sur le bouton.
- Clic sur le bouton droit de la souris (Clic D) : C'est l'opération qui permet d'activer le menu contextuel d'un élément.

### A.3 Les menus contextuels

La touche la plus utilisée sur une souris est la touche gauche, mais, une fois un élément sélectionné, en cliquant sur la touche droite de la souris pour faire apparaître un menu à côté de votre curseur, c'est ce qu'on appelle « le menu contextuel » comme il est représenté par la figure I.14. Il vous permet d'accéder directement à certaines options concernant cet objet. Pour vider la poubelle, par exemple, il suffit de cliquer avec la touche droite de votre souris sur l'icône de la poubelle de votre bureau. Il ne restera plus qu'à sélectionner l'option « Vider la poubelle » et la tâche est effectuée.



Figure I.14 : Menu contextuel sous Windows

## I.5 Introduction au système d'exploitation Unix et Linux

### I.5.1 Définition

Unix est un système d'exploitation très populaire parce qu'il est présent sur un grand nombre de plates-formes, du micro-ordinateur à l'ordinateur central (mainframe). L'avantage de cela, c'est que les programmes développés sous Unix peuvent être transférés d'une plate-forme à une autre avec un minimum de modifications.

Ce système est multitâche, c'est-à-dire qu'il est capable de gérer et d'exécuter plusieurs programmes simultanément. De plus, il est multi-utilisateurs, c'est-à-dire que plusieurs personnes peuvent s'y connecter en même temps et travailler ; le système partage alors toutes les ressources logicielles et matérielles de l'ordinateur entre les différents usagers.

L'histoire d'Unix est unique dans le monde des systèmes d'exploitation. En effet, alors que la plupart des systèmes d'exploitation ont été conçus par des fabricants d'ordinateurs pour vendre leurs machines, Unix n'a pas été conçu dans un but commercial. Il l'est devenu parce qu'il constitue une norme en matière de système d'exploitation.

Contrairement à un système d'exploitation commercial complètement contrôlé par son fabricant, le système Unix est aujourd'hui distribué par plusieurs intervenants dont voici les principaux :

- ✓ AT&T, à qui on attribue la paternité de l'Unix ;
- ✓ l'université de Berkeley, qui a fait évoluer Unix dans plusieurs domaines ;
- ✓ SUN Microsystems, à qui l'on doit les améliorations importantes de l'interface graphique ;
- ✓ Santa Cruz Operation et Microsoft, le XENIX/UNIX fut la première version pour PC de Unix.

À cause de cette situation de développement, le système d'exploitation Unix a mis plusieurs années à être standardisé. Actuellement, il en existe deux principales variantes, incompatibles entre elles :

- ✓ Unix SYSTEM V,
- ✓ Unix BSD.

Il existe aussi une multitude de variations mineures dérivées d'une des deux ou des deux principales variantes ; on a ainsi, en les regroupant :

- ✓ les systèmes Unix-Based ;
- ✓ les systèmes Unix-Like.
- ✓ Systèmes Unix-Based

Les systèmes Unix-Based ont obtenu une licence d'utilisation d'AT&T. Il s'agit d'une adaptation de Unix. Ils restent compatibles avec la version d'AT&T parce qu'ils partagent le même noyau. On retrouve dans cette catégorie :

- ✓ XENIX/UNIX provenant de SCOMicrosoft ;
- ✓ AIX provenant d'IBM ;
- ✓ Mac OsX provenant d'APPLE ;
- ✓ SunOS/Solaris provenant de SUN MicroSystems ;
- ✓ IRIX provenant de Silicon Graphics ;
- ✓ ULTRIX provenant de DIGITAL ;
- ✓ HP-UX (Hewlett Packard UniX) provenant Hewlett-Packard.

### **I.5.2 Histoire de l'UNIX**

- ✓ C'est en 1969 que Kenneth THOMPSON, employé chez Bell, développe un nouveau prototype de système à temps partagé ; son nom est Unics et sera Unix.
- ✓ Entre 1970 et 1975, Unix devient le système portable et officiel pour les institutions et les universités ;
- ✓ En 1980, des chercheurs de l'université de Berkeley développent leur propre UNIX (BSD) ;
- ✓ En 1983, la société ATT tente une commercialisation d'un UNIX système V et de concurrencer l'UNIX BSD ;
- ✓ En 1984, Richard Stallman lance le projet GNU/Linux qui vise à créer son système 'Unix' complètement libre ;
- ✓ En 1987, la société ATT finalise sa version de Unix...et donc des scissions se créent pour arriver au choix d'une licence BSD !
- ✓ En 1988, c'est le début d'une normalisation avec l'organisation 'IEEE' et la norme 'POSIX' pour les développements d'applications autour d'un système UNIX.
- ✓ En 1991, un étudiant finlandais, Linus TORVALDS, créé un noyau UNIX qui a été ajouté aux travaux du projet GNU de Stallman, a donc donné naissance à GNU/Linux.
- ✓ En 2004, un sud-africain, Mark SHUTTLEWORTH, qui est un développeur de la distribution Linux DEBIAN, lance son projet UBUNTU ; l'objectif étant de rendre populaire GNU/Linux en fournissant une interface facile à utiliser.

### **I.5.3 Histoire du Système LINUX et sa mascotte**

L'histoire du système Linux ou GNU/Linux remonte à quelques années et a impliqué de nombreux acteurs. Le projet GNU est mature et est capable de rivaliser avec les systèmes UNIX. Mais c'est en 1991 qu'un étudiant finlandais, Linus TORVALDS, diffuse le message

comme quoi il conçoit un système d'exploitation gratuit en disant que "c'est juste un hobby". Tout est parti de là car plusieurs volontaires se sont intéressés et ont fait remonter à Linus Torvalds des correctifs.

Et c'est en 1992 que naît le noyau Linux couplé au système GNU donnant le premier système d'exploitation libre. Linux a une mascotte (manchot pygmée) depuis 1996, date à laquelle de nombreux développeurs se sont accaparés le noyau linux et ont décidé que leur projet ait un logo. C'est Linus Torvald, l'initiateur du projet, qui décida que cela devait prendre l'allure chétive d'un manchot pygmée face au gros projet Unix. Et c'est Larry Ewing qui donna l'image originale du mythique 'Tux' pour quelque chose comme 'Torvalds UniX...'.

### **I.5.4 Avantage du système Linux**

Une liste non exhaustive de raisons pour installer un système Linux...

1. Installer et mettre à jour légalement et sans avoir à payer, installer la même copie de mon OS (Ubuntu) sur plusieurs machines sans me soucier des restrictions de licences ou de clés d'activation ; Distribuer des copies de mon système d'exploitation et des programmes qui tournent dessus sans violer aucune loi ;
2. Fonctionner sans utiliser d'anti-virus, de protection anti-adware ou spyware, ne pas avoir à redémarrer ma machine pendant des mois tout en recevant toujours les derniers correctifs de sécurité ;
3. Installer et exécuter facilement différentes interfaces graphiques si je n'aime pas la configuration par défaut ;
4. Installer une centaine de programmes par une simple commande ; les applications qui sont très modulaires et basées sur des bibliothèques communes en font un système extrêmement léger ; il existe des mini-distributions (Small Linux, Deli Linux,...) linux capables de s'installer sur des machines vieillissantes ;
5. Fonctionner sans avoir à 'défragmenter' mon disque dur, car Linux écrit là où il y a de la place et fait une gestion optimisée de l'espace ;
6. Avoir le contrôle total du matériel installé sur ma machine et savoir qu'il n'y a pas de porte dérobée dans mes logiciels ;

7. Pouvoir faire une énorme erreur qui nécessite la réinstallation complète de mon système et être capable de le faire en moins d'une heure, parce que j'ai mis toutes mes données sur une partition séparée du système d'exploitation et des programmes ;
8. Pouvoir démarrer mon système avec de "SUPERS" effets, aussi sympa que ceux de Windows, sur une machine qui a 3 ans... en moins de 40 secondes, temps d'identification compris (nom d'utilisateur + mot de passe),
9. Etre capable de configurer tout ce que je veux, légalement, y compris mes programmes fétiches. Je peux même contacter les développeurs du logiciel concerné pour leur poser des questions, leur donner des idées et être impliqué dans la construction ou le développement de la version en cours si j'en ai envie ;
10. Pouvoir utiliser le même matériel pendant plus de 5 ans avant qu'il n'ait réellement besoin d'être remplacé... J'ai toujours du matériel qui a presque 10 ans, qui tourne sous Linux et qui est toujours utile ;
11. Pouvoir surfer sur internet pendant que l'OS s'installe
12. Utiliser à peu près n'importe quel matériel en sachant que le pilote est déjà présent dans le système d'exploitation... éliminant ainsi la nécessité de rechercher le site du fabricant pour trouver ce pilote ;
13. Obtenir le code source de pratiquement n'importe quoi, y compris celle du noyau du système d'exploitation ou celle de la plupart de mes applications.
14. Disposer réellement d'une plus grande autonomie avec sa batterie d'ordinateur portable...
15. Lancer un plus grand nombre d'applications sans pour autant monopoliser toute la mémoire disponible ; une gestion de la mémoire virtuelle optimale qui évite un trop grand nombre de 'swaps' ;
16. Compiler soi-même son noyau et/ou ses applications afin de tirer le meilleur parti de son matériel.

### **I.5.5 Systèmes Unix-Like**

Les systèmes Unix-Like reproduisent les mêmes fonctionnalités que la version AT&T, mais le noyau du système est incompatible parce qu'il a été réécrit pour éviter le versement de droit d'auteurs à AT&T. On retrouve dans cette catégorie :

- ✓ Minix ;
- ✓ Linux ;
- ✓ FreeBSD ;
- ✓ QNX.

Cette prolifération de produits a fait apparaître un certain nombre de différences entre les systèmes, dont les principales sont :

- ✓ les communications inter-programmes ;
- ✓ la gestion de la mémoire (segmentation ou pagination) ;
- ✓ divers paramètres du système ;
- ✓ divers outils qui peuvent être intégrés dans un produit et absents dans un autre.

### I.5.6 Principales distributions de Linux

Celui qui s'intéresse aux différentes versions de Linux doit faire la différence entre le noyau du système d'exploitation proprement dit (le kernel, en anglais) et la combinaison d'utilitaires qui l'accompagnent. Les distributions se différencient par le choix du noyau et le choix des différents utilitaires disponibles. Vérifiez donc la version du noyau que vous vous procurez. Cette version figure dans de nombreuses distributions et constitue un élément commun. De nombreux utilitaires sont également communs à toutes les distributions. (Exemples : Bourne Again Shell ou l'interface graphique Xfree86). Voici une liste non exhaustive des distributions disponibles :

- ✓ CentOS
- ✓ RedHat
- ✓ Fedora
- ✓ Mandriva
- ✓ Debian
- ✓ Ubuntu
- ✓ OpenSUSE

### I.5.7 Commande descriptives de Linux

who	Afficher la liste des utilisateurs connectés
whoami	Afficher l'utilisateur de la session courante
date	Afficher la date
ps	Afficher la liste des processus de l'utilisateur
kill	Stopper un processus
passwd	Créer ou changer de mot de passe
pwd	Afficher le nom du répertoire de travail
mkdir	Créer un répertoire

cd	Changer de répertoire de travail
cat	Fusionner une liste de fichiers et afficher le résultat
head	Afficher le début d'un fichier
grep	Afficher les lignes des fichiers référencés qui contiennent une chaîne de caractères donnée
wc	Compter le nombre de mots, lignes ou caractères.
sleep	Dormir pendant un certain temps (secondes)
find	Rechercher un fichier
ls	Afficher le contenu d'un répertoire
cp	Copier un fichier dans un autre In Ajouter à un fichier existant un autre nom mv Renommer un fichier
rm	Supprimer un fichier
rmdir	Supprimer un répertoire chmod Changer les protections d'un fichier TAB. 2.1 – Commandes Posix d'Unix.
cp *	dossier (copie
mv	(move, renomme et déplace un fichier)
mv	source destination
mv *	dossier (déplace tous les fichiers du répertoire actif vers le répertoire dossier)
mkdir	(créer un répertoire)
mkdir	Répertoire
rmdir	(effacer un répertoire)
rmdir	dossier (supprime un répertoire vide)
rm	(remove, efface)
rm -R	(enlèvement récursif)
rm	Fichier
rm -i	fichier (interactivement, avec demande de confirmation)
rm -f	fichier (avec force, sans demande de confirmation)
rm -r	fichier (avec récursivité, avec les sous répertoires)
rm -rf	dossier (supprime le répertoire et tou son contenu, sans confirmation) les commandes de recherche
grep	(recherche les occurences de mots à l'intérieur de fichier)
	grep motif fichier
grep -i	motif fichier (sans tenir compte de la casse)
grep -c	motif fichier (en comptant les occurences)

grep -v	motif fichier (inverse la recherche, en excluant le "motif")
grep	expression /répertoire/fichier
grep [aFm]in	/répertoire/fichier
grep "\\$" *.txt	Les commandes d'édition
more	("pager" qui affiche page par page sans retour en arrière,
"h"	affiche l'aide contextuelle)
cat	(concatenate avec le code de fin de fichier eof=CTRL + D)
cat	fichier-un fichier-deux > fichier-un-deux
cat -n	fichier > fichier-numéroté (crée un fichier dont les lignes sont numérotés)
cat -nb	fichier (affiche sur la sortie standard les lignes numéroté, sauf les lignes vides)
head	(affiche les 10 premières lignes d'un fichier)
head -n22	fichier (affiche les 22 premières lignes)
vi	(l'éditeur en mode texte universel)
emacs	(l'éditeur GNU Emacs multi fonction pour l'édition, les mails, les news,
xemacs	la programmation, la gestion des fichiers,...) (l'éditeur GNU Emacs sous X)
diff	(différence entre deux fichiers, utiles pour chercher les modifications)
diff fichier1 fichier2	Les commandes d'impression et de conversion
lp	(la commande d'impression sur les systèmes Unix Système V)
lpr	(la commande d'impression sur les systèmes BSD et Linux)
lpr	Fichier
lpc status	(affiche l'état de la file d'attente)
lpq	(affiche les travaux d'impression et leur numéro)
lprm	(supprime un travail d'impression avec son numéro comme argument)
gv	("ghostview" permet de visualiser des fichiers POST SCRIPT)
a2ps	(convertit les fichiers ASCII en POST SCRIPT)
a2ps -4	fichier -P fichier-post-script Les commandes de compilation et d'execution
f77	compile un programme en fortran 77



### I.5.8 Structure de Linux

Le noyau gère les tâches de base du système :

- ✓ L'initialisation du système
- ✓ La gestion des ressources
- ✓ La gestion des processus
- ✓ La gestion des fichiers
- ✓ La gestion des Entrées/Sorties
- ✓ L'utilisateur communique avec le noyau par l'intermédiaire d'un SHELL.
- ✓ Les Shells sont aussi des langages de commandes et de programmation.

Les utilitaires sont des outils d'interfaçage avec le système, de programmation et de communication. Les shells les plus connus sont : BOURNE SHELL (sh) KORN-SHELL (ksh) C-SHELL TC-SHELL BOURNE AGAIN SHELL (bash)

### I.5.9 Début de session

Lorsqu'un utilisateur demande à se connecter, le système l'invite à introduire son nom d'utilisateur (code) et son mot de passe. Si ces données sont correctes, le système ouvre une session de travail et lance l'interpréteur de commandes (processus shell) qui affiche à l'écran, aussitôt après son initialisation, une invitation puis se met en attente d'ordres de l'utilisateur. Dépendamment du shell utilisé, ceci peut être un simple symbole : \$ ou bien, l'invitation peut montrer le nom de la machine.

Lorsque l'utilisateur introduit une commande, le shell vérifie si elle est correcte, puis crée, si c'est le cas, un processus qui a pour tâche d'exécuter la commande.

Par exemple la commande : `leibnitz>cp source destination` copiera le fichier source dans le fichier destination. Le shell attend la terminaison du processus créé avant d'inviter l'utilisateur à introduire une nouvelle commande.

## I.6 Liste de références bibliographiques

Le cours est confectionné en se basant sur les références bibliographiques citées ci-dessous.

- [I.1] BAUCHE FZ. « Informatique de base », polycopie de cours 1° STU, Université Djilali Bounaama-Khemis Miliana, 2018-2019.
- [I.2] Santini G. Dubacq J.-C. « Introduction à l'informatique : Cours complet de Villetaneuse S1 », IUT 2016.
- [I.3] Daoudi E. « Système d'Exploitation » Polycopié de cours I, Université de Oujda-Maroc, 2014/2015.

- [I.4] Lachachi N. « Système informatique, algorithmes et langage C », Polycopié de cours. Université Ahmed Ben Bella – Oran, 2016.
- [I.5] Duchâteau C. « Initiation à l'informatique », Cours. Université notre dame de la paix – Namur, 2002.
- [I.6] Arnaud A. « Initiation à l'informatique et à l'internet », cours pour Initiative ART&P 2012.
- [I.7] Moreau P., Morié P. « Windows XP utilisateur », Tsoft et Groupe Eyrolles, 2004.
- [I.8] « Microsoft Windows 7 » Office Professionnel de Qualification des Organismes de Formation (AFCI-NEWSOFT), , 2010.
- [I.9] Landuyt F. « Windows, une évolution unique », Promotion: M.Sc., Campus de Troyes, SUPINFO International University, 2016.
- [I.10] Lorne B. « gestion de l'ordinateur l'explorateur Windows », Février 2012.
- [I.11] Raymond J. « introduction à Windows », Service du soutien à l'enseignement, Collège de Lionel Groulx, 2009.
- [I.12] Delalin H. « Cours de systèmes d'exploitation », Département Service et Réseaux de Communication, IUT de Lens - Université d'Artois, 2005/2006.
- [I.13] Yve M. « Fenêtres et outils du système d'exploitation », Institut des sœurs du notre Dame, Anderlecht.
- [I.14] Hutzler G. « Informatique générale », Laboratoire IBISC, SITE.
- [I.15] Halfeld-Ferrari M. « Systèmes d'exploitation : Operating Systems »,
- [I.16] BOUCHERY M. « Premiers pas dans Windows 7 »
- [I.17] Luc De Mey, SITE : [www.courstechinfo.be/InitInfo/Bureau.html](http://www.courstechinfo.be/InitInfo/Bureau.html)
- [I.18] Santini G. « introduction au système d'exploitation Linux », Université de Paris, 2012.
- [I.19] FERRERE R. « Utilisation du système d'exploitation Linux et réseau », Mémoire Mathématiques appliquées - Modélisation Statistique, Module Outils informatiques, Laboratoire de Mathématiques CNRS Université de Franche-Comté, 2018.
- [I.20] Le système d'exploitation GNU-Linux/Version imprimable-Wikilivres , [https://fr.wikibooks.org/w/index.php?title=Le système d'exploitation...](https://fr.wikibooks.org/w/index.php?title=Le_système_d'exploitation...)
- [I.21] Ben Halima M. « Le système d'Exploitation Linux », Institut Supérieur d'Informatique et de Multimédia de l'université de Sfax, 2009.
- [I.22] Groupe polymtl « Chapitre 2 Introduction au système Unix/Linux », <http://www.groupe.polymtl.ca/inf2610/documentation/notes/chap2.pdf>
- [I.23] « Les commandes de base de LINUX », [http://www.lmd.jussieu.fr/~flott/polytechnique/mec583\\_08/linux\\_vi\\_f77.pdf](http://www.lmd.jussieu.fr/~flott/polytechnique/mec583_08/linux_vi_f77.pdf).
- [I.24] Casteyde C. « Guide d'installation et de configuration de Linux », Copyright © 2004 Christian Casteyde.
- [I.25] Katy Blanc, « Introduction au système GNU/LINUX ».

**CHAPITRE II : LOGICIELS ET ALGORITHMIQUE****SOMMAIRE**

II.1 Introduction .....	34
II.2 Partie software d'un système informatique .....	34
II.2.1 Types de logiciels .....	34
II.3 Langage algorithmique.....	39
II.3.1 Historique .....	40
II.3.2 Ordinateur et résolution d'un problème .....	41
II.3.3 Algo-machine .....	42
III.4 Structure d'un algorithme .....	42
III.4.1. L'en-tête d'un algorithme .....	43
III.4.2. Identificateurs .....	43
II.4.3 Variables et type des variables .....	43
II.4.4 Instruction d'affectation .....	46
II.5. Ordre des instructions .....	46
II.6. Expressions et opérateurs .....	47
II.6.1 Opérateurs numériques.....	48
II.6.2 Opérateur rationnels.....	48
II.6.3 Opérateur alphanumérique & .....	48
II.6.4 Opérateurs logiques (ou booléens) :.....	48
II.7 Tests en langage algorithmique .....	49
II.7.1 Condition et tests .....	50
II.8. Exercices avec solution .....	54
II.9 Algorithmique en biologie.....	60
II.9.1 Logiciel « PhyML » développé .....	61
II.10. Liste de références bibliographique .....	62

## II.1 Introduction

La différence entre un algorithme et un programme est souvent se traduit par l'aspect de détail. Un algorithme est souvent exprimé avec une notation indépendante de tout langage de programmation alors qu'un programme est écrit dans un langage de programmation particulier. Les langages les plus utilisés sont proches du langage courant et sont traduit de façon à ce que chaque instruction soit transmise dans une forme intelligible par la machine. Un logiciel est un programme exécutable qui utilise les ressources matérielles du système informatique. Le logiciel (software) reste toujours abstrait au matériel (hardware) qui est bien nécessaire au fonctionnement du premier où les deux sont indissociables et interdépendants.

Dans ce chapitre nous expliquons en détaille, les différents types de la partie software d'un ordinateur ainsi que les étapes et les instructions nécessaires à l'écriture d'un algorithme scientifique multidisciplinaire.

## II.2 Partie software d'un système informatique

Dans la première partie de ce chapitre, nous voudrions présenter quelques notions de base sur la partie software d'un système informatique. Comme nous le savons, qu'un ordinateur dénué de tout logiciel se trouve dans l'impossibilité de fonctionner. C'est la partie logicielle qui donne au matériel toute sa vie et son intelligence. Un logiciel est un programme de traitement de l'information contenant les procédures et les données nécessaires à une application. Le programme peut être formé par un ensemble d'instruction écrites dans des fichiers que l'ordinateur est capable d'exécuter. On en distingue deux types principales : Les systèmes d'exploitations (OS, Operating System) ou logiciel système.

### II.2.1 Types de logiciels

Suites à nos connaissances, on distingue deux types de logiciels :

- ✓ Les logiciels de base (systèmes d'exploitation). Exemples (Windows, Unix, ...). Ce type de logiciel est bien expliqué dans le chapitre I.
- ✓ Les logiciels d'applications (exécutables). Exemples (Word, Excel, MSN, ...)

#### A. Les logiciels de base

Les logiciels système sont des types de programmes informatiques conçus et nécessaires au fonctionnement d'un ordinateur et ses applications. La fonction principale de ce type de logiciel c'est qu'ils se placent entre le matériel et les autres logiciels. Ils permettent

ainsi, aux logiciels d'application une indépendance vis à vis du matériel. En plus, ils se chargent par la réalisation des fonctions de base du système (impression, visualisation, écriture mémoire ...) et enfin ils traitent les demandes d'accès au matériel des logiciels d'application. Comme exemple, on cite, les systèmes d'exploitation S.E ou O.S (Operating System), Le BIOS (Basic input output system) et les SGBD (systèmes de gestion des bases de données).

**Le système d'exploitation (voir chapitre I) :** un système d'exploitation est un logiciel de base qui assure la communication entre le processeur, les périphériques et l'utilisateur. Il est tout le premier programme qu'on peut installer sur un ordinateur. Il assure la gestion du matériel et des autres logiciels. Le système d'exploitation le plus utilisé aujourd'hui est Windows. Mais il existe d'autres comme : MSDOS, Linux, Unix, ...ect. Nous pouvons classer les systèmes d'exploitation en deux grandes familles, selon leurs modes de communication avec l'utilisateur :

- ✓ Les systèmes d'exploitation fonctionnant en mode texte comme le Ms Dos.
- ✓ Les systèmes d'exploitation fonctionnant en mode graphique comme Windows et Linux.

Le bios c'est un logiciel installé dans la mémoire morte (Read only memory) implanté sur la carte mère. Il est chargé du démarrage de notre système informatique une fois que nous allumons l'ordinateur et de la gestion du flux de données entre le système d'exploitation et les périphériques connectés tels que le disque dur, la carte vidéo, le clavier, la souris et l'imprimante.....ect.

Les fonctionnalités de base d'un système d'exploitation sont :

- ✓ Il permet à l'ordinateur de démarrer et établie une communication avec l'utilisateur
- ✓ Il se charge d'établir la communication entre le matériel et les programmes
- ✓ Il met en relation les différents périphériques connectés à l'ordinateur
- ✓ Il permet d'organiser les disques et les fichiers qu'ils contiennent
- ✓ Il gère la lecture/écriture des données sur le disque ou dans mémoire
- ✓ Il affiche les informations sur l'écran, interprète les touches tapées au clavier Etc....

## B. Logiciels d'applications

Après installation du logiciel principale qui constitué la première interface avec le matériel, les utilisateurs ont besoin d'utiliser différents types de logiciel. A cet effet, les logiciels d'applications sont conçus pour parrainer à l'OS. Les commandes effectuées par l'utilisateur dans une application sont transmises à l'OS, à son rôle les transférées au matériel, qui lui, retourne le résultat en chemin inverse. Il y a des applications ne fonctionne pas sous n'importe quelle OS sauf spécifiquement avec Microsoft Windows.

### B.1. Les logiciels utilitaires (programme utilitaire)

Il y a des programmes assurent la réalisation des fonctions complémentaires au système d'exploitation, ce sont les logiciels utilitaires. C'est un logiciel conçu pour analyser, configurer, optimiser ou entretenir une pièce d'équipement informatique, un système d'exploitation, un logiciel ou les informations enregistrées sur un support informatique. Un logiciel utilitaire est utilisé pour gérer un système informatique contrairement aux logiciels d'application, qui visent à exécuter des tâches des utilisateurs. Quelques logiciels utilitaires sont livrés avec les systèmes d'exploitation mais ils ne sont pas considérés comme faisant partie du système d'exploitation.

### B.2. Les logiciels des langages de programmation

**Ce type de logiciels peut être** considéré comme des utilitaires par ce qu'ils permettent la production d'applications. Ils sont classés de 1ère génération à 4ème génération. Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques. D'une manière générale, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire et de significations. Exemple Ex : Delphi, Fortran, Pascal, Matlab, C+, C++, Java, .....

### B.3. Les systèmes de gestion de base de données (SGBD)

On définit une base de données comme l'ensemble des données stockées d'une façon structurée et avec le moins de redondances qui modélisent une partie du monde réel pour une application informatique. Pour les manipuler, on utilise généralement un logiciel spécialisé appelé SGBD (Système de Gestion de Bases de Données) En anglais DBMS (data base management system). Ils permettent l'accès et la gestion des bases de données retrouver de l'information par son contenu en se basant sur des critères de recherche. Un SGBD c'est un outil qui permet d'insérer, modifier, retirer et rechercher des données ; le tout de façon

efficace. Les systèmes de gestion de base de données sont donc des logiciels universels, indépendants de l'usage qui est fait des bases de données. Ils sont utilisés pour de nombreuses applications informatiques, tels que :

- ✓ Commerçants (gestion des commandes,..... )
- ✓ Lieu de travail (gestion des employés, étudiants, .....)
- ✓ Hôpital, médecin (gestion des dossiers patients)
- ✓ Applications métiers (Architecte, pharmaciens, avocats, notaires.....)
- ✓ Applications web

Les SGBD sont souvent utilisés par d'autres logiciels ainsi que les administrateurs ou les développeurs. Ils peuvent être sous forme de composant logiciel, de serveur, de logiciel applicatif ou d'environnement de programmation.

Il existe de nombreux systèmes de gestion de base de données. En 2008, Oracle détenait près de la moitié du marché des SGBD avec MySQL et Oracle Database. Vient ensuite IBM avec près de 20 %, laissant peu de place pour les autres acteurs.

#### **B.4. Logiciels applicatifs (ou logiciel d'application)**

En informatique, on utilise le terme application pour désigner l'activité d'un utilisateur d'une part et le logiciel qui automatise son activité (logiciel applicatif) d'autre part. Un logiciel applicatif donc, est un ensemble de programmes informatiques nécessaires à aider l'utilisateur à réaliser un certain travail par ordinateur.

On distingue plusieurs types de ce genre de logiciels :

- ✓ Les Texteurs qui sont de logiciels de traitement de texte. Ils permettent de réaliser des documents Word (.doc), des pages web (.html ou .xml), des documents imprimés (.rtf), ... (Exemples : Word, Works, ...).
- ✓ Les Tableurs qui sont des logiciels de gestion des tableaux. Ils permettent de créer des Classeurs Excel (.XLS), des pages web (.html ou .xml), ... (Exemples : Excel, Lotus, ...).
- ✓ Les navigateurs qui sont des logiciels conçus pour consulter le World Wide Web, l'ensemble des sites web, pour lire des pages web techniquement. (Exemple : Internet explorer, Mozilla Firefox, Opera...).

- ✓ Les lecteurs multimédia qui sont des logiciels conçus pour lire les son ou vidéos de types (.MP3, .RM, .RAM, .MPEG, .WAV, .AVI, ...) Exemple : Real Player, Windows Media Player, VLC multimedia,.....
- ✓ Logiciels de traitements d'image : logiciels d'édition, de modification des images et de création des dessins. Exemples : Paint, PhotoFilter, PhotoShop, ... .
- ✓ Les logiciels de création des animations : permettent de créer des images animées (.GIF), de jeux flashes (.SWF), des pages web animés (.HTM), .... Exemples : Flash, Gimps, ... etc.
- ✓ Les logiciels de création des diaporamas : permettent de réalisé des animations destinées à être présentée devant un groupe. Ces animations sont des diaporamas de type (.ppt, .pps, ...). Exemple : Le logiciel PowerPoint.
- ✓ Les logiciels de traitement de vidéo (Windows Movie Maker, Meeve, PoweDVDCréator, ....)
- ✓ Les logiciels de jeux.
- ✓ Les logiciels de gestion des bases de données (SQL, ACSSES, ORACLE, ....).
- ✓ Les logiciels de messagerie électronique : permettent de communiquer entre deux utilisateurs. Exemples : MSN, YAHOO MESSENGER, NetMeeting, ... .
- ✓ L'antivirus : Permettent de détecter et supprimer les mauvais programmes. Exemple : Avaste, AVG, Panda, Avira, Kaspersky, ... . . . etc.

### C. Fichier Informatique

Un **fichier informatique** est une unité informationnelle physiquement stockée sur un support de mémoire de masse permanent (disque dur par exemple). Un fichier a généralement comme attributs : un nom ; un chemin d'accès ; une taille mesurée en octets ; une date de création et une de dernière modification. Plusieurs systèmes de fichiers multi-utilisateur rattachent aussi à chaque fichier un propriétaire et des droits d'accès. Enfin, chaque fichier a un contenu, soit une suite ordonnée d'octets, qui peut représenter n'importe quelle donnée binaire déterministe : un programme informatique, un document, un texte, etc.

### D. Types de logiciels

Les logiciels peuvent être classés en 4 grands types :

- ✓ Logiciels **commerciaux** vendus dans le commerce



- ✓ Logiciels misent à disposition gratuitement pour essais ou tests et qui doivent être payés pour une utilisation régulière : **Les Shareware**.
- ✓ Logiciels misent à disposition gratuitement et qui peuvent être utilisés librement sans aucune contribution : **Les Freeware**.
- ✓ Les logiciels fournis normalement avec leur code source et qui peuvent être modifiés librement : **Logiciels libres**.

### II.3 Langage algorithmique

Historiquement, plusieurs types de notations ont représenté des algorithmes. Il y a eu notamment une représentation graphique, avec des carrés, des losanges, etc. qu'on appelait des **organigrammes**. Aujourd'hui, cette représentation est quasiment abandonnée, pour deux raisons. Le premier est en relation avec la taille de l'algorithme, parce que si l'algorithme commence à grossir un peu, ce n'est plus pratique d'utiliser des organigrammes. La deuxième raison, est que cette représentation favorise le glissement vers un certain type de programmation non structurée que l'on essaye toujours de l'éviter.

A cet effet, nous utilisons généralement une série de commandes appelée « **pseudocode** », qui ressemble à un langage de programmation authentique dont on aurait évacué la plupart des problèmes de syntaxe. Ce pseudocode est différent légèrement d'un livre ou d'un enseignant à un autre.

Suivant ce qu'il est cité dans « l'Encyclopédia Universalis » nous pouvons dire que l'algorithme est une suite de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat, et cela indépendamment des données. Quelques d'autres chercheurs ont précisé que l'algorithme est un ensemble d'actions à faire, en utilisant des données initiales, pour aboutir à la solution d'un problème.

Les algorithmes ne se décrivent pas avec un langage de programmation, leur écriture ne nécessite pas un ordinateur. La résolution des différents problèmes par le langage algorithmique permettant de faciliter l'écriture d'un programme en évitant les erreurs et en économisant le temps d'exécution.

**D'où vient le nom « Algorithme »** : je peux répondre à cette question posé toujours par nos étudiants c'est que le nom « Algorithme » est provient du nom du mathématicien perse AbuAbdullah Muhammad Ibn Mussa Al-Khwarizmi (783-850 dynastie Abbasside en

(9<sup>eme</sup> siècle)), originaire de la ville de Khiva de l'ancienne région du « Khwarezm », actuellement c'est « Ouzbékistan ». Il était membre des Maisons de la sagesse et mort à Bagdad en 850. AlKhawarizmi est un chercheur très connu dans des domaines multidisciplinaires tels que les mathématiques, la géographie, l'astrologie et l'astronomie. Ses écrits, rédigés en langue arabe sont utilisés dans l'introduction de l'algèbre en Europe. Cela lui permet d'être le père de la théorie des algorithmes ainsi que de l'algèbre.

### **II.3.1 Historique**

En se basant sur nos recherches bibliographiques très vastes sur l'histoire de l'algorithme nous pouvons dire que la notion d'algorithme remonte à l'antiquité. Cela s'est précisé dans le domaine des mathématiques par l'emploi de variables par contre l'algorithme au sens informatique apparait avec l'invention des premières machines dotés d'automatismes. Les premiers algorithmes remontent à environ 1800 ans avant J.C avec les babyloniens, ensuite Euclide (PGCD) et beaucoup d'autres. Les algorithmes ne se traitent pas seulement qu'avec des nombres ; il en existe énormément qui traitent d'autres données, comme l'algorithme génétique, les algorithmes de jeux, etc.

Dans le paragraphe suivant un bref historique sur le langage algorithmique qui 'est associée toujours à la programmation informatique sera présenté en se basant sur le polycopié de cours de nos collègues de l'Université des Sciences et de la Technologie Houari Boumediene référencé par [6]. Ils ont cité un petit rappel chronologique non exhaustif d'un certain nombre d'évènements majeurs ayant contribué au développement ou carrément à la fondation de l'informatique. Du caillou au Bit ; des millénaires d'interrogations, de découvertes et d'apprentissage. Il est expliqué que depuis des milliers d'années différents outils ont été utilisé pour effectuer des différents calculs, le plus anciens est probablement l'os plus la main qui est utilisée par la plupart des sociétés. D'autres variété d'objets et de techniques seront ainsi utilisées tels que les entailles sur le bois, les entassements de cailloux et d'autres objets. Des outils plus complexes furent inventés au fur et à mesure, tel l'abaque, ensuite le boulier, la pascaline, Ci-dessous quelques évènements marquants sur l'apparition et le développement de l'informatique et de l'ordinateur :

- ✓ **780-850:** Abu Abdullah Muhammad bin Musa Al-Khawarizmi.
- ✓ **1642 :** l'invention de la Pascaline par Blaise Pascal.
- ✓ **1821 :** invention de la machine à différences par Charles Babbage.

- ✓ **1840** : Ada Lovelace (mathématicienne) nomme le processus logique d'exécution d'un programme : Algorithmes, en l'honneur à Al-Khawarizmi.
- ✓ **1854**: Boole publie "An Investigation Into the Laws of Thought", ouvrage fondateur de l'algèbre de Boole. 1946 : premier ordinateur l'ENIAC.
- ✓ **1947** : apparition du transistor (laboratoires Bell Téléphone).
- ✓ **1950** : invention de l'assembleur à l'université de Cambridge.
- ✓ **1957** : lancement du premier compilateur FORTRAN, par John Backus et son équipe (IBM).
- ✓ **1958** : Jack St. Clair Kilby invente le circuit intégré.
- ✓ **1964** : lancement de la série 360 d'ordinateurs d'IBM ; ordinateurs compatibles entre eux.
- ✓ **1968** : apparition du Langage PASCAL, créé par Niklaus Wirth.
- ✓ **1971** : commercialisation du Intel 4004 ; premier microprocesseur Intel (4 Bits, 108 KHz, 60000 instructions par seconde, composé de 2300 transistors en technologie de 10 microns).
- ✓ **1973** : lancement du mini-ordinateur multitâches (temps-réel) et multi-utilisateur, le HP 3000, par Hewlett-Packard.
- ✓ **1981** à nos jours : c'est l'âge des micro-ordinateurs et des supercalculateurs.

### II.3.2 Ordinateur et résolution d'un problème

Comme vous le savez que l'ordinateur n'est qu'une machine capable d'exécuter automatiquement une série d'opérations simples qu'on lui a demandées de faire. Donc, pour résoudre un problème à l'aide d'un ordinateur, il nous faut suivre les étapes suivantes :

1. Analyse du problème : cette étape nous permet de définir avec précision les résultats à obtenir, les informations dont on dispose, ...
2. Déterminer les méthodes de résolution: cette étape s'agit-il de déterminer la suite des opérations à effectuer pour obtenir à partir des données la solution du problème posé. Cette suite d'opérations constitue un **algorithme**. Nous pouvons souvent de fournir plusieurs algorithmes pour un seul problème, il faudra juste de choisir le plus efficace.
3. Formuler l'algorithme définitif: cette étape doit faciliter la résolution sur ordinateur par l'expression de l'algorithme dans un formalisme adéquat (langage de description d'algorithme: LDA, organigramme, arbre programmatique, ...).
4. Traduire l'algorithme dans un langage de programmation adapté pour obtenir un programme complet qu'on peut l'écrire sur l'ordinateur.

### II.3.3 Algo-machine

L'exécution des algorithmes est très facile et se trouve dans notre vie quotidienne. Si nous utilisons un livre de recettes de cuisine pour suivre les étapes de cuisson d'un plat, ou si nous déchiffrons un mode d'emploi traduit directement du chinois pour faire fonctionner un microscope ou un distillateur ou, si nous avons indiqué un chemin à un touriste égaré donc on peut dire que nous avons déjà exécuté des algorithmes.

Donc, nous pouvons conclure comme nous avons cité auparavant qu'un algorithme, est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné. Si l'algorithme est juste, le résultat est le résultat voulu par contre si l'algorithme est faux, le résultat est, disons, aléatoire. L'algorithme doit donc contenir uniquement des instructions compréhensibles par celui qui devra l'exécuter pour le faire fonctionner. C'est d'ailleurs l'un des points forts pour les concepteurs de modes d'emploi par rapport à la différence culturelle ou lexicales, des utilisateurs, étant variables, un même mode d'emploi peut être très clair pour certains et difficile à comprendre pour d'autres. En informatique, il n'y a pas ce problème parce que les ordinateurs sont les récepteurs auxquelles on doit donner des instructions.

Pour maîtriser l'algorithmique vous devez avoir deux qualités, très complémentaires :

- ✓ il faut avoir une certaine anticipation, par ce qu'aucun catalogue ne permet de savoir a priori quelles instructions permettront d'obtenir le résultat voulu.
- ✓ il faut être méthodique et rigoureux : à chaque fois vous écrivez une série d'instructions que vous croyez justes, il faut systématiquement se mettre à la place de l'ordinateur qui va les exécuter, mais en utilisant un papier et un stylo pour vérifier si le résultat obtenu est bien celui que vous voulez. Cette étape reste indispensable, si vous ne voulez pas écrire des instructions ambiguës et à force de pratique, vous pouvez sauter cette dernière étape. Mais en réalité, cet apprentissage demande beaucoup de temps de travail sérieux. Aussi, dans un premier temps, vous devez éviter de sauter les étapes : la vérification méthodique, pas à pas, de chacun de nos algorithmes représente plus de la moitié du travail à accomplir.

### III.4 Structure d'un algorithme

La présentation d'un algorithme sous forme de bloc est très proche du langage de programmation. La structure générale d'un algorithme est la suivante :

**{En-tête}**

**Algorithme** <identificateur\_nom > ;

**{Partie déclarations des variables d'entrées et de sorties}****Déclaration des variables d'entrées et de sorties**

<Identificateur> : <Type de variable > ;

**{Corps de l'algorithme}****Début**

<partie actions > ;

**Fin.****III.4.1. L'en-tête d'un algorithme**

L'en-tête d'un algorithme se constitue le mot clé « **Algorithme** » indiquant le début d'un algorithme (en Pascal on utilise le mot Program) avec **l'identificateur\_nom** qui est un nom choisi par le programmeur. Généralement, on choisit un nom indiquant le rôle de l'algorithme. Notons que le point-virgule (;) indique la fin de l'en-tête. Il permet de séparer l'en-tête du reste de l'algorithme. Elle contient la déclaration de tous les objets manipulés par un algorithme tels que les constantes et les variables. Elle associe à chaque objet un nom, un type et une valeur pour les constantes.

**III.4.2. Identificateurs**

C'est un nom que l'on attribue à toute entité manipulée dans un programme. Les identificateurs sont choisis librement, par l'utilisateur, mais il doit respecter certaines règles:

1. Un nom doit commencer par une lettre et non par un chiffre par exemple : g1 ou G1 et non 1G.
2. L'identificateur doit être constitué uniquement de lettres, de chiffres et du soulignement (éviter les caractères de ponctuation et les espaces) par exemple : SNV2022, UDBKM et non SNV 2022,U D B K M .
3. L'identificateur doit être différent des mots clés réservés au langage (par exemple en Pascal: Var, begin, *sqrt*, *write* ..).

**II.4.3 Variables et type des variables**

Dans un programme informatique, on a besoin en permanence de stocker provisoirement des valeurs issues du disque dur, fournies par l'utilisateur (frappées au clavier), ou d'autre. Il

peut aussi s'agir de résultats obtenus par le programme, intermédiaires ou définitifs. Ces données peuvent être trouvés sous forme de plusieurs types : des nombres, du texte, ou autre. Toujours est-il que dès que l'on a besoin de stocker une information au cours d'un programme, on utilise une variable. On peut considérer une variable comme une boîte, que le programme va retrouver par une étiquette. Pour avoir accès au contenu de la boîte, il suffit de faire un appel à son étiquette. En réalité, il n'y a pas une vraie boîte, et pas davantage de vraie étiquette collée dessus. Dans l'ordinateur, physiquement, il y a un emplacement de mémoire, repéré par une adresse binaire. Si on programmait dans un langage directement compréhensible par la machine, Nous devons nous abstenir d'identifier nos données par de superbes codes binaires. Ce type de langages s'appelle assembleur mais ils ne sont pas les seuls langages disponibles.

### A. Déclaration des variables

Avant d'utiliser la variable, vous devez créer la boîte et de lui coller une étiquette. Cet effet doit se faire au début de l'algorithme. C'est ce qu'on appelle la **déclaration des variables**.

Le nom de la variable ou l'étiquette de la boîte obéit à des impératifs changeant selon les langages. Le nom de la variable peut comporter des lettres et des chiffres, mais qu'il exclut la plupart des signes de ponctuation, en particulier les espaces. Pour donner un nom correct à la variable vous devez le commencer par une lettre. Quant au nombre maximal de signes pour un nom de variable, il dépend du langage utilisé. En langage algorithmique, vous êtes libre de du nombre de signes pour un nom de variable.

Lorsque vous déclarez une variable, la création d'une boîte où vous réserver un emplacement mémoire ne suffit pas; il faut encore préciser ce que vous voudrez mettre dedans, car de cela dépendent la **taille** de l'emplacement mémoire et le **type de codage** utilisé.

### B. Types d'une variable

#### B.1. Types numériques classiques

Ce type est réservé à la variable destinée à recevoir des nombres. Si un octet est réservé pour coder un nombre, vous ne pouvez coder que  $2^8 = 256$  valeurs différentes. Cela peut signifier par exemple les nombres entiers de 1 à 256, ou de 0 à 255, ou de  $-127$  à  $+128$ ... Si deux octets sont réservés, vous avez droit à 65 536 valeurs; avec trois octets, 16 777 216, etc. Le

type de codage ou le type de variable choisi pour un nombre va déterminer les valeurs maximales et minimales des nombres pouvant être stockés dans la variable ainsi que la précision dans le cas de nombres décimaux. Certains langages autorisent d'autres types numériques, tels que : le type **monétaire** (avec strictement deux chiffres après la virgule) et le type **date** (jour/mois/année).

En langage algorithmique le pseudo-code utilisé pour une déclaration de variables est :

### Variable G1 en Numérique

**ou encore**

**Variables Moy, Note, PrixTTC en Numérique**

## B.2. Type alphanumérique

Les boîtes que sont les variables peuvent contenir d'autres informations que les nombres. Si par exemple on veut stocker un nom de famille, on doit disposer donc également le **type alphanumérique** appelé aussi **type caractère**, ou **chaîne de caractères** (En anglais, le type string). Dans ce type de variable, on peut stocker des **caractères** (de lettres, de signes de ponctuation, d'espaces, ou même de chiffres). Le nombre maximal de caractères pouvant être stockés dans une seule variable dépend du langage utilisé. Un groupe de caractères qu'il soit ou non stocké dans une variable, est donc souvent appelé **chaîne de caractères** (**En pseudo-code, une chaîne de caractères est toujours notée entre guillemets**). A cet effet, il faut éviter deux sources principales de possibles confusions :

- Confusion entre des nombres et des suites de chiffres. Par exemple, 243 peut représenter le nombre 243 ou la suite de caractères 2, 4, et 3. Et ce n'est pas du tout la même chose. Avec le premier, on peut faire des calculs, avec le second, on ne peut pas. Ici, on doit utiliser les guillemets permettent d'éviter toute ambiguïté : s'il n'y en a pas, 243 est deux cent quarante-trois par contre "243" représente la suite des chiffres 2, 4, 3.
- Confusion qui consiste à se mélanger les pinces entre le nom d'une variable et son contenu. Pour parler simplement, cela consiste à confondre l'étiquette d'une boîte et ce qu'il y a à l'intérieur.

## B.3 Type booléen

Dans le type **booléen** de variables on peut stocker seulement les deux valeurs logiques **VRAI** et **FAUX** (En anglais TRUE et FALSE) ou des nombres (0 et 1). Il faut comprendre que le type booléen est très économique en termes de place mémoire occupée, puisque pour stocker une telle information binaire, un seul bit suffit.

## II.4.4 Instruction d'affectation

La variable n'est pas un outil difficile à gérer, le programmeur peut ne pas faire plusieurs opérations avec une variable, mais seulement une et une seule. Cette seule opération que le programmeur puisse faire avec une variable, c'est **l'affecter**, c'est-à-dire **lui attribuer une valeur**. En pseudocode, l'instruction d'affectation se note avec le signe  $\leftarrow$

Par exemple **D1  $\leftarrow$  24** c.à.d. : Attribue la valeur 24 à la variable **D1**. On peut dire que **D1** soit une variable de type numérique. Si **D1** a été défini dans un autre type, il faut bien comprendre que cette instruction provoquera une erreur. On peut aussi attribuer la valeur d'une autre variable à une variable, telle quelle ou modifiée. Par exemple **D2  $\leftarrow$  D1** signifie que la valeur de **D2** est maintenant celle de **D1**. il faut noter qu'une instruction d'affectation ne modifie que ce qui est situé à gauche de la flèche. Par exemple : **D2  $\leftarrow$  D1 + 4**

Si **D1** contenait 12, **D2** vaut maintenant 16. De même que précédemment.

Il faut faire attention à ne pas oublier les guillemets lors d'une affectation de chaîne, ce qui se trouve à droite du signe d'affectation ne correspond à aucune variable précédemment déclarée et affectée. L'oubli des guillemets donne une erreur d'exécution.

## II.5. Ordre des instructions

L'ordre dans lequel les instructions sont écrites joue un rôle essentiel dans le résultat final d'exécution. Considérons les deux algorithmes suivants :

### Exemple 1

**Variable A en Numérique**

**Début**

A  $\leftarrow$  20

A  $\leftarrow$  10

**Fin**

### Exemple 2

**Variable A en Numérique**

**Début**

A  $\leftarrow$  10

A  $\leftarrow$  20

**Fin**



Dans le premier exemple la valeur finale de A est 10, dans l'autre elle est 20. Il n'y a aucun intérêt à affecter une variable pour l'affecter différemment juste après. En l'occurrence, on aurait tout aussi bien atteint le même résultat en écrivant simplement :

**Exemple 1****Variable A en Numérique****Début**

A ← 10

**Fin****Exemple 2****Variable A en Numérique****Début**

A ← 20

**Fin**

## II.6. Expressions et opérateurs

Dans une instruction d'affectation, à gauche de la flèche, il y a un nom de variable, et uniquement cela. Si on voit à gauche d'une flèche d'affectation autre chose qu'un nom de variable, on peut être certain qu'il s'agit d'une erreur. Par contre, à droite de la flèche, on peut trouver ce qu'on appelle une **expression qui est un ensemble de valeurs, reliées par des opérateurs, et équivalent à une seule valeur**. Par exemple, voyons quelques expressions de type **numérique** :

6

4+3

113-35+834

D1-12+5-D2

Elles sont toutes des expressions valides, pour peu que D1 et D2 soient bien des nombres. Car dans le cas contraire, la quatrième expression n'a pas de sens. Donc, les opérateurs que nous avons employés sont l'addition (+) et la soustraction (-). Il faut que l'expression située à droite de la flèche soit du même type que la variable située à gauche. D'où, **un opérateur est un signe qui relie deux valeurs, pour produire un résultat**. Les opérateurs possibles dépendent du type des valeurs qui sont en jeu.

### II.6.1 Opérateurs numériques

Ce sont les quatre opérations arithmétiques classique : + : l'addition ; - : la soustraction ; \* : la multiplication ; / : la Division. Il y a aussi, le ^ qui signifie « puissance ». **25 au carré** s'écrira donc **25 ^ 2**.

De préférence d'utiliser les parenthèses, avec les mêmes règles qu'en mathématiques. La multiplication et la division ont « naturellement » priorité sur l'addition et la soustraction. Les parenthèses ne sont ainsi utiles que pour modifier cette priorité naturelle. Cela signifie qu'en informatique,  $10 * 3 + 4$  et  $(10 * 3) + 4$  valent strictement la même chose, à savoir **34**. En revanche,  $10 * (3 + 4)$  vaut  $10 * 7$  soit 70.

### II.6.2 Opérateur rationnels

Les opérateurs rationnels: ils permettent de comparer deux valeurs de même type (numérique ou caractère) en fournissant un résultat booléen (vrai ou faux) tels que : > ; < ; ≤ ; ≥ ; = ; <> (différent) .

#### Exemple :

$C \leftarrow A > B$ . la variable C doit être de type booléen. A et B sont de même type.

### II.6.3 Opérateur alphanumérique &

Cet opérateur permet de **concaténer** (agglomérer) deux chaînes de caractères. Par exemple :

**Variables A, B, C en Caractère**

**Début**

$A \leftarrow "L2"$

$B \leftarrow "SNV"$

$C \leftarrow A \& B$

**Fin**

La valeur de C à la fin de l'algorithme est "L2SNV".

### II.6.4 Opérateurs logiques (ou booléens) :

Il s'agit du ET, du OU, du NON logique.

**N.B**

1- Il faut faire attention à la similitude de vocabulaire entre les mathématiques et l'informatique.

✓ **En mathématiques** : une « variable » est généralement une inconnue, qui recouvre un nombre non précisé de valeurs. Lorsque j'écris :

$y = 3x + 2$  ; les « variables »  $x$  et  $y$  satisfaisant à l'équation existent en nombre infini (graphiquement, l'ensemble des solutions à cette équation dessine une droite). Lorsque j'écris :

$ax^2 + bx + c = 0$  ; la « variable »  $x$  désigne les solutions à cette équation, c'est-à-dire zéro, une ou deux valeurs à la fois...

✓ **En informatique** : une variable possède à un moment donné une valeur et une seule. Elle peut ne pas avoir de valeur du tout (une fois qu'elle a été déclarée, et n'est pas affectée). Mais ce qui est important, c'est que cette valeur justement, ne « varie » pas. Du moins ne varie-t-elle que lorsqu'elle est l'objet d'une instruction d'affectation.

2- En algorithmique le signe d'affectation est  $\leftarrow$ . Mais en pratique, la quasi-totalité des langages emploient le signe égal  $=$ . En maths,  $A = B$  et  $B = A$  sont deux propositions strictement équivalentes. En informatique, absolument pas, puisque cela revient à écrire  $A \leftarrow B$  et  $B \leftarrow A$ , deux choses bien différentes. De même,  $A = A + 1$ , qui en mathématiques, constitue une équation sans solution, représente en programmation une action tout à fait licite.

## II.7 Tests en langage algorithmique

Les structures alternatives désignent tous problèmes offrant que deux issues possibles. Il n'y a que **deux formes possibles de structure alternative (conditionnelles)** pour un test ; la première est la plus simple (réduite), la seconde la plus complexe (complète).

### 1° forme : Réduite

**Si** booléen **Alors**

Instructions

**Finsi**

**Si non ne rien faire**

### 2° forme : complète

**Si** booléen **Alors**

Instructions 1

**Sinon**

Instructions 2

**Finsi**

Un **booléen** est une **expression** dont la valeur est **VRAIE** ou **FAUX**. Cela peut donc être (il n'y a que deux possibilités) : une **variable** (ou une expression) de type booléen et une **condition**. Dans la forme la plus simple (**Si... Alors**) la machine examine la valeur du booléen. Si ce booléen a pour valeur **VRAI**, elle exécute la série d'instructions. Cette série d'instructions peut être très brève comme très longue, cela n'a aucune importance. En revanche, dans le cas où le booléen est **FAUX**, l'ordinateur ne rien faire et saute directement aux instructions situées après le FinSi.

Dans le cas de la structure complète, si le booléen est **VRAI**, et après avoir exécuté la série d'instructions 1, au moment où elle arrive au mot « Sinon », la machine saute directement à la première instruction située après le « Finsi ». De même, au cas où le booléen a comme valeur **Faux**, la machine saute directement à la première ligne située après le « Sinon » et exécute l'ensemble des « instructions 2 ». Dans tous les cas, les instructions situées juste après le FinSi seront exécutées normalement.

### II.7.1 Condition et tests

La condition est composée de trois éléments : la valeur ; l'opérateur de comparaison et l'autre valeur. Les valeurs peuvent être de n'importe quel type (numériques, caractères...). Mais pour une comparaison correcte, il faut que les deux valeurs de la comparaison soient du même type. Les **opérateurs de comparaison** sont : égal à = ; différent de < > ; strictement plus petit que < ; strictement plus grand que > ; plus petit ou égal à < = ; plus grand ou égal à > = . À noter que ces opérateurs de comparaison peuvent employer avec des caractères. Ceux-ci sont codés par la machine dans l'ordre alphabétique, les majuscules étant systématiquement placées avant les minuscules.

#### A. Conditions composées

Certains problèmes exigent parfois de formuler des conditions composés. Par exemple si « la variable **C** est inclus entre 3 et 15 ». Cette phrase comporte **deux** conditions : « **C** est supérieur à 3 et **C** est inférieur à **15** ». Il y a donc bien là deux conditions, reliées par un **Opérateur Logique** c'est le mot **ET**. En informatique il y a quatre opérateurs logiques : **ET, OU, NON, et XOR**.

✓ Le **ET** : Pour que "Condition1 ET Condition2" soit VRAI, il faut impérativement que Condition1 soit VRAI et que Condition2 soit VRAI. Dans tous les autres cas, "**Condition 1 et Condition 2**" sera faux.

✓ Le **OU** : Pour que "Condition1 OU Condition2" soit VRAI, il suffit que Condition1 soit VRAIE ou que Condition2 soit VRAIE. Le point important est que si Condition1 est VRAIE et que Condition2 est VRAIE aussi, Condition1 OU Condition2 reste VRAIE.

✓ **Le XOR** (ou OU exclusif) : Pour que "Condition1 XOR Condition2" soit VRAI, il faut que soit Condition1 soit VRAI, soit que Condition2 soit VRAI. Si toutes les deux sont fausses, ou que toutes les deux sont VRAI, alors le résultat global est considéré comme FAUX.

✓ **le NON** inverse une condition : NON(Condition1) est VRAI si Condition1 est FAUX, et il sera FAUX si Condition1 est VRAI. C'est l'équivalent pour les booléens du signe "moins" que l'on place devant les nombres. Tous les opérateurs logiques précédents sont représentés dans des **tables de vérité** (C1 et C2 représentent deux conditions) :

C1 <b>ET</b> C2	C2 Vrai	C2 Faux
C1 Vrai	Vrai	Faux
C1 Faux	Faux	Faux

C1 <b>ou</b> C2	C2 Vrai	C2 Faux
C1 Vrai	Vrai	Vrai
C1 Faux	Vrai	Faux

C1 <b>XOR</b> C2	C2 Vrai	C2 Faux
C1 Vrai	Faux	Vrai
C1 Faux	Vrai	Faux

<b>NON</b> C1	
C1 Vrai	Faux
C1 Faux	Vrai

## B. Tests imbriqués

Ce type de tests est nécessaire dans le cas où deux conditions ne suffisent pas. Par exemple, un programme devant donner l'état de l'eau selon sa température doit pouvoir choisir entre trois réponses possibles (solide, liquide ou gazeuse). La première solution possible est la suivante :

**Variable** Temp en Entier

**Début**

**Ecrire** "Entrez la température de l'eau :"

**Lire** Temp

**Si** Temp  $\leq$  0 **Alors**

**Ecrire** "C'est de la glace"

**FinSi**

**Si** Temp > 0 **Et** Temp < 100 **Alors**

**Ecrire** "C'est du liquide"

**Finsi**

**Si** Temp > 100 **Alors**

**Ecrire** "C'est de la vapeur"

**Finsi**

**Fin**

Dans ce cas, les conditions se ressemblent plus ou moins et la machine doit examiner trois tests successifs alors que tous portent sur une même chose, la température de l'eau (la valeur de la variable Temp). Il serait ainsi bien plus rationnel d'**imbriquer** les tests de cette manière:

**Variable Temp en Entier**

**Début**

**Ecrire** "Entrez la température de l'eau :"

**Lire** Temp

**Si** Temp  $\leq$  0 **Alors**

**Ecrire** "C'est de la glace"

**Sinon**

**Si** Temp < 100 **Alors**

**Ecrire** "C'est du liquide"

**Sinon**

**Ecrire** "C'est de la vapeur"

**Finsi**

**Finsi**

**Fin**

Donc, l'algorithme est réduit et au lieu de devoir taper trois conditions composées, nous n'avons plus que deux conditions simples. Cette méthode est avantageuse pour économiser le temps d'exécution de l'ordinateur. Si la température est inférieure à zéro, celui-ci écrit dorénavant « C'est de la glace » et passe **directement** à la fin, sans être ralenti par l'examen

d'autres possibilités (qui sont forcément fausses). On peut considérer donc que les structures de tests imbriqués sont donc un outil indispensable à la simplification et à l'optimisation des algorithmes.

### C. Utilisation des variables Booléennes

Pour écrire les tests, nous avons utilisé uniquement des **conditions**. Mais, il existe un type de variables (les booléennes) responsable au stockage des valeurs **VRAI** ou **FAUX**. On peut donc entrer des conditions dans ces variables, et tester ensuite la valeur de ces variables.

Reprenons l'exemple de l'eau. On pourrait le réécrire comme suit :

#### Variable Temp en Entier

#### Variables A, B en Booléen

#### Début

**Ecrire** "Entrez la température de l'eau :"

**Lire** Temp

A ← Temp ≤ 0

B ← Temp < 100

#### Si A Alors

**Ecrire** "C'est de la glace"

#### SinonSi B Alors

**Ecrire** "C'est du liquide"

#### Sinon

**Ecrire** "C'est de la vapeur"

#### Finsi

#### Fin

Avec cette technique, il apparaît qu'on a alourdi plutôt qu'allégé l'algorithme de départ, en ayant recours à deux variables supplémentaires. Mais il faut prendre en compte qu'une variable booléenne n'a besoin qu'un seul bit pour être stockée. De ce point de vue, l'alourdissement n'est donc pas considérable. Alors, dans certains cas, notamment celui de conditions composées très lourdes (avec plein de ET et de OU tout partout) cette technique peut faciliter le travail du programmeur, en améliorant nettement la lisibilité de l'algorithme.

Dans le cas de conditions composées, les parenthèses jouent un rôle fondamental.

**Variables A, B, C, D, E en Booléen**

**Variable X en Entier**

**Début**

**Lire X**

$A \leftarrow X > 12$

$B \leftarrow X > 2$

$C \leftarrow X < 6$

$D \leftarrow (A \text{ ET } B) \text{ OU } C$

$E \leftarrow A \text{ ET } (B \text{ OU } C)$

**Ecrire D, E**

**Fin**

Si  $X = 3$ , alors on remarque que D sera VRAI alors que E sera FAUX.

S'il n'y a dans une condition que des ET, ou que des OU, en revanche, les parenthèses ne changent strictement rien. Dans une condition composée employant à la fois des opérateurs ET et des opérateurs OU, la présence de parenthèses possède une influence sur le résultat, tout comme dans le cas d'une expression numérique comportant des multiplications et des additions.

## II.8. Exercices avec solution

Pour bien comprendre ce qu'on a expliqué dans les paragraphes précédents, nous proposons à nos étudiants dans la suite, des exercices corrigés confectionnés e plusieurs supports pédagogiques publiés dans plusieurs universités (voir références bibliographiques à la fin de ce chapitre)

**Exercice 1 :** Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on laisse de côté le cas où le nombre vaut zéro).

**Exercice 2 :** Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit **pas** calculer le produit des deux nombres.

**Exercice 3 :** Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).

**Exercice 4 :** Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !



**Exercice 5 :** Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :

- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" après 12 ans

Peut-on concevoir plusieurs algorithmes équivalents menant à ce résultat ?

**Exercice 6 :** Formulez un algorithme équivalent à l'algorithme suivant :

**Si** Tutu > Toto + 4 **OU** Tata = "OK" **Alors**

Tutu ← Tutu + 1

**Sinon**

Tutu ← Tutu - 1

**Finsi**

**Exercice 7 :** Il lira au clavier l'heure et les minutes, et il affichera l'heure qu'il sera une minute plus tard. Par exemple, si l'utilisateur tape 21 puis 32, l'algorithme doit répondre :

"Dans une minute, il sera 21 heure(s) 33".

NB : on suppose que l'utilisateur entre une heure valide. Pas besoin donc de la vérifier.

**Exercice 8 :** De même que le précédent, cet algorithme doit demander une heure et en afficher une autre. Mais cette fois, il doit gérer également les secondes, et afficher l'heure qu'il sera une seconde plus tard.

Par exemple, si l'utilisateur tape 21, puis 32, puis 8, l'algorithme doit répondre : "Dans une seconde, il sera 21 heure(s), 32 minute(s) et 9 seconde(s)".

NB : là encore, on suppose que l'utilisateur entre une date valide.

**Exercice 9 :** Un magasin de reprographie facture 10 DA les dix premières photocopies, 9 DA les vingt suivantes et 8 DA au-delà. Ecrivez un algorithme qui demande à l'utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

**Exercice 10 :** Les habitants de Paris paient l'impôt selon les règles suivantes :

- les hommes de plus de 20 ans paient l'impôt
- les femmes paient l'impôt si elles ont entre 18 et 35 ans
- les autres ne paient pas d'impôt

Le programme demandera donc l'âge et le sexe du Paris, et se prononcera donc ensuite sur le fait que l'habitant est imposable.

## **CORRIGES DES EXERCICES**

### **Corrigé de l'exercice 1**

**Variable n en Entier**

**Début**

**Ecrire** "Entrez un nombre : "

**Lire** n

**Si**  $n > 0$  **Alors**

**Ecrire** "Ce nombre est positif"

**Sinon**

**Ecrire** "Ce nombre est négatif"

**Finsi**

**Fin**

### **Corrigé de l'exercice 2**

**Variables m, n en Entier**

**Début**

**Ecrire** "Entrez deux nombres : "

**Lire** m, n

**Si**  $(m > 0 \text{ ET } n > 0)$  **OU**  $(m < 0 \text{ ET } n < 0)$  **Alors**

**Ecrire** "Leur produit est positif"

**Sinon**

**Ecrire** "Leur produit est négatif"

**Finsi**

**Fin**

### **Corrigé de l'exercice 3**

**Variable n en Entier**

**Début**

**Ecrire** "Entrez un nombre : "

**Lire** n

**Si**  $n < 0$  **Alors**

**Ecrire** "Ce nombre est négatif"

**SinonSi**  $n = 0$  **Alors**

**Ecrire** "Ce nombre est nul"

**Sinon**

**Ecrire** "Ce nombre est positif"

**Finsi**

**Fin**

#### Corrigé de l'exercice 4

**Variables**  $m, n$  **en Entier**

**Début**

**Ecrire** "Entrez deux nombres : "

**Lire**  $m, n$

**Si**  $m = 0$  **OU**  $n = 0$  **Alors**

**Ecrire** "Le produit est nul"

**SinonSi**  $(m < 0$  **ET**  $n < 0)$  **OU**  $(m > 0$  **ET**  $n > 0)$  **Alors**

**Ecrire** "Le produit est positif"

**Sinon**

**Ecrire** "Le produit est négatif"

**Finsi**

**Fin**

Si on souhaite simplifier l'écriture de la condition lourde du **SinonSi**, on peut toujours passer par des variables booléennes intermédiaires.

#### Corrigé de l'exercice 5

**Variable**  $age$  **en Entier**

**Début**

**Ecrire** "Entrez l'âge de l'enfant : "

**Lire**  $age$

**Si**  $age \geq 12$  **Alors**

**Ecrire** "Catégorie Cadet"

**SinonSi**  $age \geq 10$  **Alors**

**Ecrire** "Catégorie Minimale"

**SinonSi** age  $\geq$  8 Alors

**Ecrire** "Catégorie Pupille"

**SinonSi** age  $\geq$  6 Alors

**Ecrire** "Catégorie Poussin"

**Finsi**

**Fin**

### Corrigé de l'exercice 6

Aucune difficulté, il suffit d'appliquer la règle de la transformation du OU en ET vue en cours (loi de Morgan). Attention toutefois à la rigueur dans la transformation des conditions en leur contraire...

**Si** Tutu  $\leq$  Toto + 4 ET Tata  $\neq$  "OK" **Alors**

    Tutu  $\leftarrow$  Tutu - 1

**Sinon**

    Tutu  $\leftarrow$  Tutu + 1

**Finsi**

### Corrigé de l'exercice 7

**Variables** h, m **en Numérique**

**Début**

**Ecrire** "Entrez les heures, puis les minutes : "

**Lire** h, m

m  $\leftarrow$  m + 1

**Si** m = 60 **Alors**

    m  $\leftarrow$  0

    h  $\leftarrow$  h + 1

**FinSi**

**Si** h = 24 **Alors**

    h  $\leftarrow$  0

**FinSi**

**Ecrire** "Dans une minute il sera ", h, "heure(s) ", m, "minute(s)"

**Fin**

Corrigé de l'exercice 8

Variables h, m, s en Numérique

**Début**

**Ecrire** "Entrez les heures, puis les minutes, puis les secondes : "

**Lire** h, m, s

$s \leftarrow s + 1$

**Si**  $s = 60$  **Alors**

$s \leftarrow 0$

$m \leftarrow m + 1$

**FinSi**

**Si**  $m = 60$  **Alors**

$m \leftarrow 0$

$h \leftarrow h + 1$

**FinSi**

**Si**  $h = 24$  **Alors**

$h \leftarrow 0$

**FinSi**

**Ecrire** "Dans une seconde il sera ", h, "h", m, "m et ", s, "s"

**Fin**

Corrigé de l'exercice 9

Variables n, p en Numérique

**Début**

**Ecrire** "Nombre de photocopies : "

**Lire** n

**Si**  $n \leq 10$  **Alors**

$p \leftarrow n * 0,1$

**SinonSi**  $n \leq 30$  **Alors**

$p \leftarrow 10 * 0,1 + (n - 10) * 0,09$

**Sinon**

$p \leftarrow 10 * 0,1 + 20 * 0,09 + (n - 30) * 0,08$

**FinSi**

**Ecrire** "Le prix total est: ", p

**Fin**

### Corrigé de l'exercice 10

**Variable sex en Caractère**

**Variable age en Numérique**

**Variabes C1, C2 en Booléen**

**Début**

**Ecrire** "Entrez le sexe (M/F) : "

**Lire** sex

**Ecrire** "Entrez l'âge: "

**Lire** age

C1 ← sex = "M" ET age > 20

C2 ← sex = "F" ET (age > 18 ET age < 35)

**Si** C1 ou C2 **Alors**

**Ecrire** "Imposable"

**Sinon**

**Ecrire** "Non Imposable"

**FinSi**

**Fin**

## II.9 Algorithmique en biologie

Cette partie du cours indique à nos étudiants de la deuxième année du domaine sciences de la nature et de la vie que le langage algorithmique pourra être utilisé pour résoudre des problèmes qui se posent dans leurs spécialités qui se basent sur la biologie générale. Ce type d'algorithmes génétiques peut donner aux étudiants et aux chercheurs une solution très utile pour savoir quel est le degré de parenté et l'histoire évolutive entre les différentes espèces de **levures** ou de vers, ou d'**insectes**, ou d'oiseaux qu'ils ont en train d'étudier durant leurs cursus. Ce type d'algorithme génétiques appartient à la famille des algorithmes évolutionnistes utilise la notion de sélection naturelle et l'appliquent à une population de solutions potentielles au problème donné qui ne possède pas d'une solution connu.

La méthode la plus fiable qui n'induit pas en erreur par des ressemblances morphologiques superficielles, c'est la phylogénie moléculaire : comparer le génome de bestioles. Elle se base sur la disponibilité d'une banque de données regroupant des millions de séquences d'ADN correspondant à des dizaines de milliers d'espèces vivantes. **Sans outils adaptés pour les comparer, elles resteront une suite muette de A, T, C et G.**

« Sebastián Escalón » de futura sciences a déclaré que « Olivier Gascuel » et « Stéphane Guindon » chercheurs au Laboratoire d'informatique, de robotique et de microélectronique de Montpellier (Lirmm) ont mis au point en 2003 un puissant algorithme permettant d'estimer les relations évolutives liant un ensemble d'organismes à partir de la comparaison de leur ADN ou bien des acides aminés de leurs protéines. Cette méthode a eu un succès important, tant et si bien que le papier qui les présentait dans Systematic Biology est aujourd'hui en bonne place dans le Science Citation Index.

### II.9.1 Logiciel « PhyML » développé

L'utilisation de l'algorithme ADN, permet de réduire le temps de calcul par rapport aux méthodes classiques. De plus, il peut être utilisé avec des données bien plus complexes et longues. Le chercheur « Olivier Gascuel » a cité qu'il y a des recherches que l'on n'imaginait pas pouvoir faire et qui ne demandent maintenant que quelques minutes grâce à cet algorithme comme. Sur la base de cet algorithme, les chercheurs ont développé un logiciel baptisé s'appelle « **PhyML** » que je conseil mes étudiants de l'apprendre par ce qu'il est en accès libre sur Internet. Ils peuvent profiter de sa simplicité d'utilisation comme les chercheurs du monde entier. Il suffit juste d'entrer les données et d'attendre que les huit processeurs de l'ordinateur du laboratoire « Lirmm » s'en occupent. Ils reçoivent les résultats par e-mail quelques heures ou minutes plus tard. Il y a trois ou quatre utilisateurs en permanence s'occupe du serveur qui tourne en permanence comme il a indiqué « Olivier Gascuel ». Le principe de base de l'algorithme est celui du maximum de vraisemblance, concept majeur en statistique. Il consiste ici à définir un modèle hypothétique qui décrit à la fois le degré de parenté entre les espèces étudiées et les mutations qui ont pu se produire à partir de la séquence originale postulée. Ensuite, il calcule la probabilité que les données correspondent à l'hypothèse de départ. Le processus est réitéré, modèle après modèle, jusqu'à ce que cette probabilité soit maximale. Le modèle correspondant constitue la réponse de l'algorithme.

Avec le logiciel « PhyML », vous pouvez voir les espèces former un arbre où chaque nœud correspond à un ancêtre commun et où la longueur des branches représente le temps pendant lequel les espèces ont évolué. C'est grâce à ce logiciel, ils ont pu reconstruire

l'arbre phylogénétique des primates à partir de séquences d'ADN représentant plus de 900000 paires de bases. Chaque nœud correspond à un ancêtre commun, et la longueur des branches, au temps d'évolution. Maintenant, les chercheurs peuvent comprendre les processus évolutifs au niveau des génomes. Un vaste programme de recherches auquel participent les deux chercheurs « Olivier Gascuel » s'intéresse aux séquences répétées de l'ADN tandis que « Stéphane Guindon » travaille sur les mécanismes de sélection des mutations chez les virus pour développer des nouvelles versions du programme.

## **II.10. Liste de références bibliographique**

- [II.1] BAUCHE FZ. « Informatique de base », polycopie de cours 1° STU, Université Djilali Bounaama-Khemis Miliana, 2018-2019.
- [II.2] Menu N., « Informatique Matériel Et Logiciel Chapitre 2 Hardware & Software », 2000 / DGEO2.
- [II.3] Santini G., Dubacq J.-C., « Introduction à l'informatique : Cours complet de Villetaneuse S1 », IUT-2016.
- [II.4] TORCHE Y. « Chapitre 3 : Les logiciels et algorithmique ». Polycopie de cours, Université de Mila. 2020.
- [II.5] DJEBLI H. « Introduction l'Administration de Bases de Données-Chapitre 1 : Rappels sur les BD relationnelles », Licence 3 en Sciences de l'Informatique, université Constantine 2, 2016/2017.
- [II.6] Rabia AMOUR, Rachid FERMOUS, Mohand AIT OUARABI, Moufida BENZEKKA, Smain YOUNSI. « Cours, Exercices et Programmation Pascal Première Année Universitaire Domaines : SM, ST et MI ALGORITHMIQUE ». Université des Sciences et de la Technologie Houari Boumediene. 2017-2018.
- [II.7] AMIAR L. « Méthodes Algorithmiques », Polycopié de cours, Université Larbi Ben M'hidi – Oum El Bouaghi. 2018-2019.
- [II.8] Darmangeat C. « Cours d'algorithmique et de programmation ». Support de cours. Université Paris.
- [II.9] <https://fr.scribd.com/document/14871334/ALGORITHMIQUE>
- [II.10] BOUDRIES A. « Algorithmique Cours et travaux dirigés ». Polycopié de cours. Université Abderrahmane MIRA de Béjaia. Année 2016.



- [II.11] GHAZI S. « Algorithmique et structures de donnée Avancées ». Polycopie du Cours. Université Badji Mokhtar Annaba. Année universitaire 2020-2021.
- [II.12] Filliâtre J-C. « INF411 Les bases de la programmation et de l'algorithmique ». support de cours. École Polytechnique - Édition 2019.
- [II.13] Benoit A, Depardon B., Moulleron C., Resvoy C. « Algorithmique I ». Cours et Travaux Dirigés L3, Ecole Normale Supérieure de Lyon. 2008-2009.
- [II.14] DJEFFAL A. « Cours d'Algorithmique et structures de données », université Mohamed Khider – Biskra. 2012-2013.
- [II.15] Cormen T.H., Leiserson C.E., Rivest R.L., Cormen T.H., and Cormen T.H. “Introduction à l’algorithmique ». Dunod, 1996.
- [II.16] Courtin J. and Kowarski I. ”Initiation à l’algorithmique et aux structures de données ». Dunod, 1990.
- [II.17] Hoarau P., « langage algorithmique »
- [II.18] Darmangeat C. « Cours d'algorithmique et de programmation », du master MECI, l'université paris 7.
- [II.19] Marie-agnès peraldi-frati « Cours algorithmique et programmation informatique », uns/iut de nice côte d’azur.
- [II.20] Knuth d.e. « algorithmes », csl publications 201.
- [II.21] Dowek G. ed rpa. « introductipon a la science informatique » 2010.
- [II.22] Julliand Ed Vuibert. « cours et exercices corrigés d’algorithmique- j », fev 2010.
- [II.23] Gascuel O. « L'algorithmie qui fait avancer la biologie », Lirmm, Montpellier.  
[olivier.gascuel@lirmm.fr](mailto:olivier.gascuel@lirmm.fr)
- [II.24] <https://www.futura-sciences.com/tech/actualites/tech-algorithme-fait-avancer-biologie/>

## **CONCLUSION GENERALE**

La science de l'informatique est devenue la plus pertinente pour trouver la solution de plusieurs problèmes dans les différents secteurs et domaines multidisciplinaires. Dans ce polycopié de cours destiné aux étudiants de la deuxième année tronc commun du domaine SNV, nous avons présenté les notions de base de l'informatique en ce qui concerne les systèmes d'exploitation très utilisés tels que Windows, l'UNIX et LINUX ainsi que les notions de bases de la partie software d'un ordinateur et les différents aspects d'écriture d'un algorithme.

Le système d'exploitation est le premier logiciel principal nécessaire au démarrage d'un ordinateur. Les trois systèmes d'exploitation les plus utilisés pour les ordinateurs personnels sont Windows, Unix et Linux. Pour cela, nous avons examiné dans le premier chapitre : la structure globale et les mécanismes de base de ces trois systèmes. Dans le chapitre suivant, nous avons essayé de présenter les types de la partie logiciel d'un ordinateur et les principales instructions de base de conception d'un algorithme qui est un outil nécessaire permettant aux étudiants de se préparer à écrire des programmes répondant à leurs besoins.

Ce polycopié de cours sera un support largement utilisé dans divers spécialités du tronc commun du domaine SNV. Il fournit un ensemble de fonctionnalités qui facilitent l'utilisation d'un système d'exploitation, ainsi que les logiciels d'application d'un ordinateur et enfin l'écriture des algorithmes nécessaires à la programmation informatique.