



# ETUDE DU SYSTEME D'EXPLOITATION MS DOS 6.22

Version 2

**Ecole Internationale des Technologies Informatiques Avancées**

**Direction Pédagogique**

27, Rue de Palestine, Bettana .Salé. Maroc

[www .eitia .ma](http://www.eitia.ma) Email : [eitia@eitia.ma](mailto:eitia@eitia.ma)

Tél. : 88 2000 / 88 2010 Fax. : 88 2010

## SOMMAIRE

Sommaire.....	2
CHAP 1 NOTIONS PRELIMINAIRES .....	4
I. Définitions Et Généralités: .....	4
1. Définition de l'informatique:.....	4
2. Les multiples du bit .....	4
3. Notions de « SOFTWARE » et de « HARDWARE ».....	4
4. Les opérations réalisables par un ordinateur .....	5
5. Codage de l'information.....	6
6. Notion de programmation.....	8
II. SCHEMA DE PRINCIPAL D'UN MICRO-ORDINATEUR.....	8
1. La mémoire centrale : .....	9
2. L'unité de traitement : .....	9
3. Les Bus.....	10
4. Les interface d'Entrée / Sortie.....	10
III. NOTION DE SYSTEME D'EXPLOITATION.....	10
1. Introduction.....	10
2. Rôle d'un système d'exploitation.....	10
3. les principaux types de systèmes d'exploitation .....	11
CHAP 2 PRESENTATION GENERALE DU SYSTEME MS-DOS .....	12
I. Historique.....	12
II. STRUCTURE DU SYSTEME MS-DOS .....	13
1. Le noyau.....	14
2. Les utilitaires : .....	14
III. PROCEDURE DU DEMARRAGE D'UN PC.....	15
IV. FORME GENERALE D'UNE LIGNE DE COMMANDE.....	17
CHAP 3 : GESTION DES SUPPORTS MAGNETIQUES.....	19
I. NOTIONS PRELIMINAIRES.....	19
1. Unité Logique : .....	19
2. Partitionnement d'un disque dur .....	19
3. Unité de démarrage ou de lancement.....	19
4. Interpréteur de commandes.....	19
5. Notions de commande externe et de commande interne.....	20
6. Activation d'une unité.....	20
7. Notion de disque virtuel.....	20

II.	PRI NCIPALES COMMANDES RELATIVES AUX DI SQUES.....	21
1.	Formatage ou initialisation d'une disquette.....	21
2.	La commande LABEL.....	22
3.	La commande VOL.....	22
4.	La commande DiskCopy.....	22
5.	la commande DiskComp.....	22
6.	La commande ChkDsk.....	23
7.	la commande SUBST.....	23
CHAP 4 :	GESTI ON DES REPERTOI RES.....	24
I.	DEFINI TIONS ET GENERALI TES.....	24
1.	Définition d'un répertoire.....	24
2.	Utilité des répertoires.....	24
3.	Organisation logique d'un disque.....	24
4.	Répertoires particuliers.....	25
5.	Chemin d'accès.....	25
II.	PRI NCIPALES COMMANDES RELATIVES AUX REPERTOI RES.....	26
1.	Création d'un nouveau répertoire MD ou Mkdir.....	26
2.	Activation d'un répertoire CD ou ChDir.....	26
3.	Suppression d'un répertoire vide RD ou Rmdir.....	26
4.	Affichage d'une arborescence Tree.....	26
5.	Spécification de plusieurs chemins d'accès aux fichiers exécutables Path.....	26
6.	Fonctionnement :.....	27
7.	Intérêt de la commande :.....	27
CHAP 5 :	OPERATI ONS SUR LES FI CHI ERS.....	28
I.	NOTI ONS COMPLEMENTAI RES.....	28
1.	Les caractères génériques.....	28
2.	Les fichiers spéciaux.....	28
3.	Les redirections d'entrée/Sortie.....	29
4.	Enchaînement de commandes.....	30
5.	Notion de filtre.....	30
II.	ETUDE S DES PRI NCIPALES COMMANDES RELATIVES AUX FI CHI ERS.....	31
1.	Manipulation des attributs d'un fichier.....	31
2.	Lecture d'un fichier texte.....	32
3.	Duplication de fichiers : la commande COPY.....	32
4.	Copie de fichiers et de sous-répertoires.....	33
5.	Changement de noms de fichiers : La commande REN.....	34

# CHAP 1 NOTIONS PRELIMINAIRES

## I. DEFINITIONS ET GENERALITES:

### 1. Définition de l'informatique:

C'est la science qui étudie le traitement automatique et rationnel de l'information.

Informatique = support de connaissance

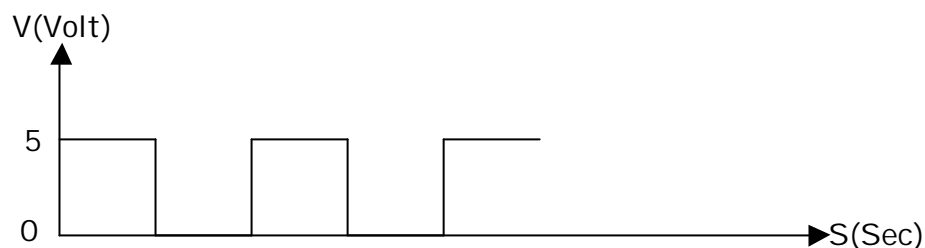
Unité de l'information

L'information est une grandeur qui peut être confiée. L'unité fondamentale qui sert à l'exprimer est le **bit** (Binary Digit).

Le bit constitue l'atome de l'information. Il ne peut prendre que deux valeurs possibles: **0** ou **1**. Ces deux chiffres constituent l'alphabet d'un système de numération très utilisé en informatique: c'est le système binaire. Pour cette raison, on les appelle aussi chiffres binaires (ou binary digits).

Justification du bit comme unité fondamentale de l'information est essentiellement dû au fait que les systèmes physiques de traitement et de stockage de l'information sont des systèmes à deux états stables:

#### Exemple:



V peut représenter la d.d.p. aux bornes d'une capacité (mémoire élémentaire).

Dans ce cas :

- l'état 1 ----> capacité chargée,
- l'état 0 ---> capacité déchargée .

Pour cette raison, ces systèmes sont aussi, souvent appelés **systèmes logiques** ou **digitaux**.

### 2. Les multiples du bit

Les multiples du bit les plus utilisés dans la pratique sont les suivants :

L'octet (ou byte) = 8 bits ,

Le Kilo Octet (Ko) =  $2^{10}$  o = 1024 o

Le Méga Octet (Mo) =  $2^{20}$  o = 1024 Ko

Le Giga Octet (Go) =  $2^{30}$  o = 1024 Mo

### 3. Notions de « SOFTWARE » et de « HARDWARE »

Dans le domaine de l'informatique, on peut distingué principalement deux aspects différents :

- \* L'aspect matériel (ou HARDWARE) qui concerne l'étude de la technologie et de l'architecture des systèmes informatiques ,
- \* L'aspect logiciel (ou SOFTWARE) qui concerne l'étude des techniques d'exploitation, de programmation et d'utilisation des ordinateurs.

#### 4. Les opérations réalisables par un ordinateur

Un ordinateur est une machine capable d'effectuer avec une grande vitesse et une probabilité d'erreur nulle les opérations suivantes :

##### A. Opérations arithmétiques :

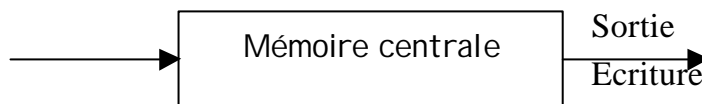
- \* Addition (+),
- \* Soustraction (-),
- \* Multiplication (\*),
- \* Division (/).

Remarque : En réalité, ces opérations se ramènent toutes à deux opérations de base : l'addition et le décalage (voir cours d'architecture).

##### B. Opérations d'entrée/sortie :

Ce sont des opérations qui concernent la communication des informations entre la mémoire centrale de l'ordinateur et son extérieur :

- \* Une opération d'entrée ou de lecture est un transfert d'informations de l'extérieur vers la mémoire centrale de l'ordinateur.
- \* Généralement, les informations d'entrée constituent les programmes de traitement ainsi que les données à traiter, et les informations de sortie les résultats d'un traitement :



Remarque :

L'extérieur de l'ordinateur est représenté par les organes périphériques ou dispositifs d'entrée/sortie (Ecran, Clavier, Imprimante, Disquette, Souris, etc.). Et la communication avec tous ces dispositifs physiques est assurée par le système d'exploitation. Ceci décharge l'utilisateur final de la programmation de ces tâches souvent non évidente à réaliser.

##### C. Opérations logiques :

Elles sont essentiellement basées sur les trois opérateurs suivants :

- le OU logique (**OR**),
- le E logique (**END**),
- la négation logique (**AND**).

Exemple d'expression logique :  $(X > 0) \text{ AND } (X < 20)$ .

Le résultat d'une telle expression, évalué à l'aide de l'unité arithmétique et logique (UAL), ne peut prendre que deux valeurs possible : **vrai** (true) ou **faux** (false).

### **D. Opérations de décision :**

Ce type d'opérations consiste en réalité en la possibilité de n'exécuter une tâche (une ou plusieurs instructions) que si une condition est réalisée ; ou de répéter une même tâche si une condition de répétition est vérifiée. Il joue un rôle très important dans le domaine de l'algorithmique.

La condition sur laquelle se base la décision d'exécuter la tâche ou de la répéter se présente toujours sous la forme d'une expression à évaluer et dont le résultat permet de fixer la décision à prendre.

Exemple :

#### **Répéter**

- Lire (X) ;
- $C = (X < 0) \text{ OU } (X > 20)$  ;
- **Si** (C= Faux) **Alors** afficher message d'erreur ;

**Jusqu' à (C= Vrai).**

## **5. Codage de l'information**

### **A. Nécessité d'un codage numérique**

Un examen rapide des opérations par un ordinateur montre que ces opérations portent essentiellement sur des grandeurs numériques. De plus, à l'intérieur de machine les informations sont représentées sous forme électrique (absence ou présence de tension ou de courant électrique).

Par conséquent, le traitement d'une information quelconque par ordinateur nécessite tout d'abord de la transformer sous forme de nombre binaires : c'est le principe du codage numérique.

### **B. Principaux codes numériques :**

Pour traduire numériquement une information, différents codes universellement connus peuvent être utilisés. Ces codes sont en général proposés par des organismes de normalisation nationaux (ANSI, AFNOR, etc.). Et homologués par des organismes internationaux (ISO, CCI TT, etc.).

On peut citer, à titre d'exemples (voir annexes), les codes :

- \* BINNAIRE PUR,
- \* BAUDOT (alphabet télégraphique international utilisé en TELEX).
- \* TELETEX (Avis T.61 du CCI TT),
- \* EBCDIC,
- \* CODAR-U/FD (Norme ISO 29),
- \* ASCII (Norme ISO 646),
- \* Etc.

Mais, le plus connu et le plus utilisé dans la pratique, notamment en micro-informatique, est le code ASCII (American Standard Code for Information Interchange) qui a été

proposé à l'origine par l'ANSI ( American National Standards Institute) puis homologué par l'ISO (International Standards Organisation).

**C. Principales caractéristiques du code ASCII**

Chaque caractère ou symbole utilisé couramment par l'homme pour écrire des informations est codé à l'aide d'une combinaison de 8 bits (c à d 1 octet). Cette combinaison représente un nombre exprimé dans la base 2 (nombre binaire) et qui peut être traduit ou transformé dans n'importe quelle autre base (voir cours d'architecture).

Exemples :

<b>Symbole</b>	<b>ASCII</b>	<b>Décimal</b>
A	01000001	65
a	01100001	97
Z	01011010	90
+	00101011	43
?	00111111	63
o	01101111	111
O	01001111	97
0	00110000	48

- Le nombre total de combinaison est :  $2^8 = 256$  combinaisons . Mais seules les 128 première de 0 à 127) sont standards . Les 128 combinaisons restantes ( de 128 à 255) peuvent changer d'une machine à une autre ou même pour une même machine (voir notion de **pages de code**).

- Les codes des lettres de l'alphabet naturel et ceux des chiffres décimaux sont choisis de façon à faciliter les opérations de classement ou de tri d'information qui se font essentiellement par comparaison de code ASCII .

Pour cela, les codes affectés à ces symboles préservent leur ordre naturel.

Ainsi :

- \* Les lettres minuscules a, b, c,.....z sont codées respectivement par les nombres 97,98,99 ,.....,122 ;
- \* Les lettres majuscules A, B , C,.....,Z ont pour codes respectifs 65, 66, 67,.....,99 ;
- \* Et les chiffres décimaux 0, 1, 2,.....,9 sont codés de 48 à 57.

- Les 32 premiers codes (de 0 à 31) sont des codes spéciaux utilisés pour désigner des caractères non affichables ou des commandes de contrôle :

Exemples :

<b>Code</b>	<b>Signification</b>
13	Carriage Return (CR)
7	Sonnerie (Bell)
3	Fin de texte (ETX)
4	Fin de Transmission (EOT)

## 6. Notion de programmation

### A. Nécessité de programmation

L'ordinateur ne pouvant effectuer qu'un nombre limité d'opérations, la résolution d'un problème quelconque par ordinateur nécessite :

- \* de disposer d'une méthode de résolution (ou algorithme).
- \* De décomposer cette méthode sous la forme d'une série d'opérations élémentaires réalisables par la machine : c'est le principe de la programmation .

### B. Notion d'instruction

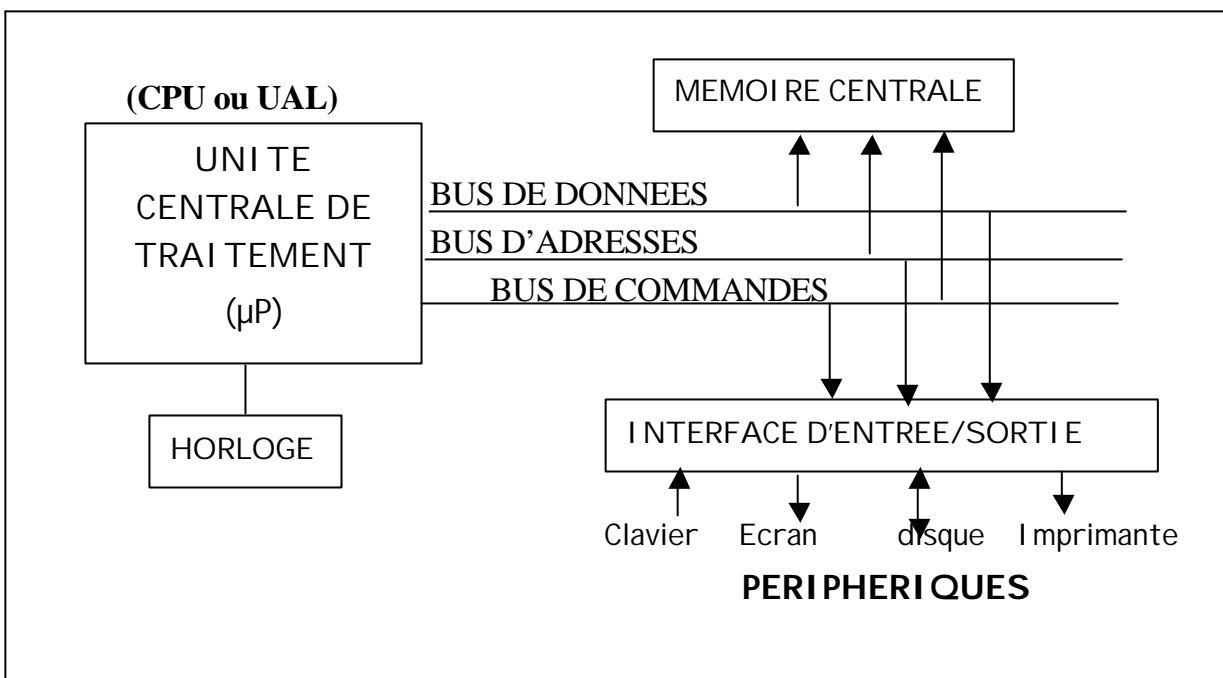
C'est l'opération élémentaire réalisable par la machine et présentée sous la forme d'une commande à exécuter, et qu'il faut exprimer dans un langage de programmation donné.

### C. Notion de programme :

C'est un ensemble d'instructions dont l'exécution par la machine fournit la solution d'un problème bien déterminé.

Dans la pratique, lorsque la taille d'un programme informatique est relativement importante, on parle plutôt de **logiciel** . De même lorsqu'un logiciel intègre plusieurs fonctions différents (traitement de texte + tableurs + gestionnaire de base de données, etc. .) on le désigne souvent par le terme **progiciel**.

## II. SCHEMA DE PRINCIPE D'UN MICRO-ORDINATEUR



L'examen de ce schéma, très simplifié , permet de constater la présence de trois blocs fonctionnels principaux :

- \* La mémoire centrale (MC)
- \* L'unité centrale de traitement (CPU ou UAL).



- \* Les cartes d'interface d'Entrée/Sortie.

Chacun de ces blocs est constitué en réalité d'un ensemble plus ou moins complexe de circuit électroniques intégrés (CI). L'étude détaillée de chacun de ces éléments étant à faire dans le cadre du cours « Architecture des Ordinateurs », dans ce qui suit nous allons nous contenter d'en donner les définitions de base ainsi que les informations qui nous seront utiles pour la suite du cours « Système d'Exploitation ».

### 1. La mémoire centrale :

Elle est constituée d'un ensemble de circuits capable de :

- \* recevoir,
- \* stocker,
- \* et restituer

Toute sorte d'informations (programmes de traitement, données à traiter, résultats de traitement, etc.).

Ces circuits peuvent être divisé en deux grandes catégories :

- \* Les RAM.
- \* Les ROM.

#### A. Mémoires RAM (*Random Access Memory*)

Ce sont des mémoires accessibles en lecture et en écriture. Elles se caractérisent principalement par le fait qu'elles perdent leurs contenu dès qu'elles ne sont plus alimentées en courant électronique. De ce fait, on les appelle aussi des mémoires **vives** ou **volatiles**.

#### B. Mémoire ROM (*Read Only Memory*)

Ce sont des mémoires qui ne peuvent être utilisé qu'en **lecture** . Elles se caractérisent par le fait qu'elles gardent leurs contenu de façon permanente.

Pour cela, elles sont aussi appelées des **mémoires mortes**.

Le chargement de ce type de mémoire en informations se fait au moment de leur construction moyennant des procédés spécifiques. L'effacement de ces informations est possible, en les exposant par exemple à des **rayons UV** pendant une certaine durée.

#### Remarque :

Tout micro-ordinateur doit disposer au moins d'une mémoire morte : c'est la ROM BIOS qui contient le programme de démarrage de l'ordinateur (voir chapitre suivant).

### 2. L'unité de traitement :

L'unité centrale (UCT, en anglais CPU), ou microprocesseur, exécute les instructions de programme et effectue des calculs mathématiques et logiques. Les coprocesseurs mathématiques sont des microprocesseurs qui ont été optimisés pour les calculs mathématiques complexes. La plupart des unités centrales modernes réunissent les deux périphériques dans un seul progiciel.

Les processeurs de PC sont généralement des processeurs Intel (par exemple, 8088, 8086, 80286, 80386, 80486, Pentium, Pentium II, ou Pentium III), ou d'un fabricant différent qui emploie l'ensemble des instructions établies par Intel, ces processeurs étant donc compatibles avec Intel (par exemple, AMD, Cyrix).

Les microprocesseurs permet aussi de contrôler les circuits mémoires ainsi que les différents dispositifs d'Entrée/Sortie. Son **travail** est cadencé par une **horloge** à quartz **dont la fréquence fixe la vitesse de traitement** du micro-ordinateur.

L'horloge synchronise le fonctionnement des différents composants du PC et fournit à l'unité centrale le signal de synchronisation élémentaire. L'horloge est composée d'un générateur d'horloge qui crée un signal de synchronisation, et de multiplicateurs d'horloge qui prennent ce signal de synchronisation et y appliquent un facteur de multiplication pour produire différents signaux de synchronisation pour différents composants.

### **3. Les Bus**

---> Ils constituent le support de transfert d'informations entre le microprocesseur et les différents autres circuits.

---> Chaque bus sert à transmettre un seul bit d'information,

---> Le bus le plus important est le bus d'adresse en raison de son étroite relation avec l'espace mémoire maximum utilisable par la machine.

### **4. Les interface d'Entrée / Sortie**

Ce sont des circuits d'adaptation dont le rôle principal consiste à transformer les signaux issus d'un périphérique donné en vue de les rendre utilisables par l'unité centrale et inversement.

## **III. NOTION DE SYSTEME D'EXPLOITATION**

---

### **1. Introduction**

Comme nous l'avons déjà signalé au début de ce chapitre, un ordinateur permet d'effectuer un nombre très réduit d'opérations élémentaires. Cela veut dire tout simplement que n'importe quelle tâche à réaliser par ordinateur doit d'abord être programmée et confirmée !.notamment, les supports magnétiques ainsi que plupart des dispositifs d'entrée-sortie.

Pour pallier à cette difficulté, qui pourrait avoir comme conséquence directe une sévère limitation de l'utilisation de l'informatique, est toujours fourni avec un certain nombre de programmes pré-établis soit par le constructeur ou par une société tierce. Ces programmes constituent ce qu'on appelle le système d'exploitation de l'ordinateur.

### **2. Rôle d'un système d'exploitation**

C'est de décharger l'utilisateur final de la programmation de tâches :

- routinières,
- indispensables,
- et surtout difficiles à programmer .

#### **exemples :**

- gestion de la mémoire centrale,
- gestion des supports magnétiques(ou mémoires de masse),disquette, disque dur
- contrôle des périphérique d'entrée - sortie,
- ....etc.

### 3. les principaux types de systèmes d'exploitation

#### A. systèmes multitâches

Ce sont des systèmes qui permettent l'exécution simultanée de plusieurs programmes.

Multitâche==> traitement en parallèle.

#### B. Systèmes Multi-Utilisateurs

Ils permettent le partage des ressources (matérielles et logicielles) d'un même ordinateur entre plusieurs utilisateurs simultanés.

#### C. Système Mono tâche

Ce sont des systèmes qui ne permettent l'exécution que d'un seul programme à la fois.

#### D. Système Mono-utilisateur

Ce sont les systèmes les plus simples et les plus utilisés à l'échelle de la micro-informatique. Ils se caractérisent par le fait qu'ils ne permettent pas le partage des ressources d'une même machine par plusieurs utilisateurs à la fois.

**Exemples :** MS- DOS, Unix, Windows 9X, Windows 2000, Windows XP, linux

#### **Remarque :**

- Un système multitâche n'est pas nécessairement Multi-Utilisateur ,
- Un système mono tâche n'est pas nécessairement mono-utilisateur, c'est le cas , par exemple, des premiers systèmes utilisés avec les grands ordinateurs.

# CHAP 2 PRESENTATION GENERALE DU SYSTEME MS-DOS

## I. HISTORIQUE

---

Le système **DOS** (**Disk Operating Système**) a été conçu à l'origine par Seattle Computer puis racheté par **Microsoft** en 1980, juste avant l'apparition du premier ordinateur de type **IBM PC**.

Partant de ce produit, Microsoft a annoncé sa première version du système MS-DOS (Version 1.0) en 1981.

Cette version répondait aux besoins des premiers micro-ordinateurs PC d'IBM dont les performances étaient très modestes comparées à celles des PC des années 90.

### Exemples :

- Taille de la RAM < 64 Ko,
- Disque dur : absent,
- Lecteur de disquette : simple face (capacité = 160 Ko).

Depuis cette date, le système MS-DOS a dû subir plusieurs améliorations et modifications afin de suivre l'évolution rapide et des performances de toute la gamme des micro-ordinateurs IBM PC & Compatibles.

Les principales étapes de cette évolution peuvent être résumées comme suit :

- 1983 : Version 2.0

Cette version était destinée à être utilisée sur des PC/XT dont les principales caractéristiques sont :

- RAM : 160 Ko,
- Disque Dur : 10 Mo,
- Lecteur de disquettes : 5,25 pouces, double face (360 Ko),
- 1984 : version 3.0

Cette version a coïncidé avec l'apparition des PC/AT qui ont apporté principalement les performances suivantes :

- RAM : 640 Ko,
- Lecteur de disquettes : 5,25 pouces, Haute Densité (1,2 Mo),
- Capacité de stockage du Disque dur : > 20 Mo.

- **1985 : Version 3.3**

Les principales caractéristiques de cette version sont :

- Reconnaissance des lecteurs de disquettes 3,5 pouces (720 Ko ou 1,44 Mo).
- Possibilité de sélection de pages de codes de caractères spécifiques à chaque pays (ASCII étendu).

C'est une version très stable, et qui est encore très utilisée jusqu'à présent par de nombreux programmeurs. Le seul inconvénient qu'elle présente réside dans le fait qu'elle ne permettait d'exploiter la totalité d'un disque dur dont la capacité de stockage dépasse **32 Mo** qu'en le partitionnant en plusieurs disques logiques ( grâce à l'utilitaire **FDISK**).

- **1989 : version 4.0**

Cette version a porté la taille maximale d'une partition de disque dur à 2 Go, permettant ainsi de dépasser largement la limite de 32 Mo imposée par la version 3.3.

Cependant, son utilisation n'a pas connu beaucoup de succès. Et ce ne raison de nombreuses anomalies qu'elle comportait. Ceci explique pourquoi il n'est pas exclu de trouver actuellement dans le commerce des logiciels qui ne tournent pas sous cette version .

- **1991 : version 5.0**

Ces principales caractéristiques sont :

- Une meilleure gestion de la mémoire centrale.
- Un noyau plus réduit (réduction d'environ 5 Ko)
- Possibilité de charger des gestionnaires de périphériques en mémoire haute (cf. chapitre suivant).

- **1992 : version 6.0**

Avec cette version on a vu, pour la première fois dans le système MS-DOS, l'apparition de trois nouveaux types de programmes utilitaires :

- un Antivirus (MSAV)
- un programme de compression de données (DbISpace).
- Et un utilitaire pour l'optimisation de l'utilisation de la mémoire centrale (MemMaker).

- **Juin 1993 : Version 6.20**

C'est une version apporté des corrections à des certains erreurs de la version précédente, ainsi qu'un nouveau utilitaire (DoubleGuard) destiné à contrôler le fonctionnement des programmes DbISpace et ScanDisk (qui corrige les erreurs éventuelles de disques).

Au cours de l'année 1994, et suite à une poursuite judiciaire de MicroSoft par la société Satck Electronic (pour avoir utilisé illégalement son algorithme de compression de données dans le programme DbISpace), une nouvelle version du système MS-DOS est apparue : il s'agit de la version 6021 qui ne contient plus l'utilitaire DbISpace.

- **Juillet 1994 : version 6.22**

Cette version a permis tout simplement la ré-introduction, sous une nouvelle forme, du programme de compression de données DbISpace. Elle est la dernière connue actuellement au Maroc (Janvier 1995).

## **II. STRUCTURE DU SYSTEME MS-DOS**

---

Comme tout système d'exploitation, MS-DOS est constitué de deux parties bien distinctes : Le noyau, et Les programmes utilitaires.

## 1. Le noyau

IL regroupe les routines indispensables au fonctionnement de la machine .Son chargement en RAM se fait de façon automatique au démarrage de l'ordinateur grâce au programme de lancement qui réside dans la ROM BIOS.

Ces routines peuvent être divisées en deux grandes parties :

- une partie résidente
- une partie transitoire.

La partie résidente demeure en RAM durant toute une session de travail ; c'est à dire jusqu'à ce que la machine soit mise hors tension.

La partie transitoire peut, si nécessaire, libérer provisoirement l'espace mémoire su 'elle occupe au profil d'un programme en cours d'exécution. Ceci peut être utile lorsqu'on veut exécuter un programme de grande taille ou qui traite une grande quantité d'informations.

Dans ce cas, la fin du programme entraîne automatiquement le chargement en RAM de cette partie qui sera recherché dans l'unité de démarrage (voir TP)

Le schéma suivant donne une représentation simplifiée de la configuration de la RAM conventionnelle d'un PC :

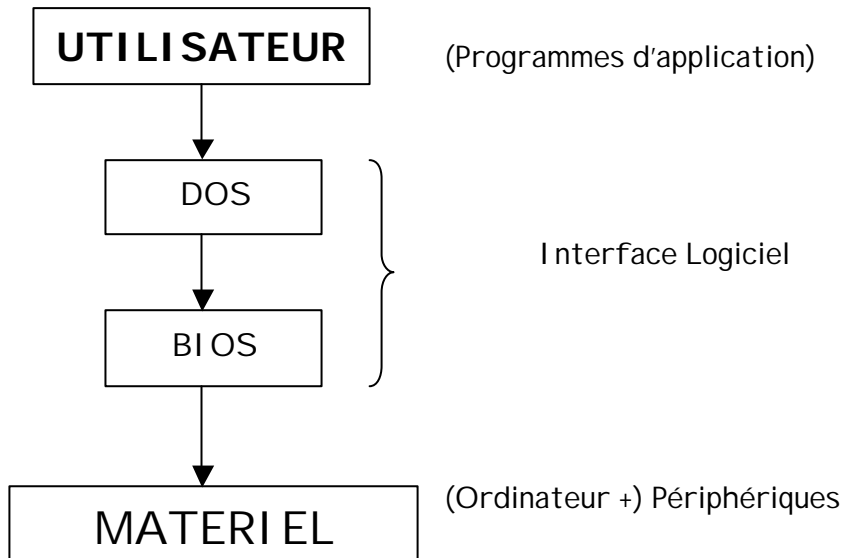
Partie Transitoire du <b>DOS</b>
Zone réservée aux Données et aux Programmes de l'utilisateur
Partie réservée du <b>DOS</b>
Zone réservée au <b>Bios</b>
Zone réservée aux <b>Vecteurs d'interruptions.</b>

## 2. Les utilitaires :

Ce sont des programmes non indispensables au fonctionnement de la machine. Ils sont comparable à n'importe quel programme exécutable écrit et mis au point par l'utilisateur.

Leur principal rôle consiste à former une interface logiciel entre l'utilisateur final et le matériel, permettant ainsi à l'utilisateur d'exploiter les ressources matérielles de la configuration utilisée sans avoir à les programmer directement. Ce qui est, nous l'avons déjà dit, non évident à faire pour tous les utilisateurs !.

Le schéma simplifié suivant montre la nécessité de passer par cette interface logicielle pour pouvoir utiliser les ressources matérielles de la configuration utilisée.



### III. PROCEDURE DU DEMARRAGE D'UN PC

Cette procédure regroupe toutes les opérations qui s'effectuent de façon automatique, c'est à dire sans l'intervention de l'utilisateur, dès la mise sous tension de la machine et jusqu'à ce que le système devienne prêt pour recevoir et interpréter les commandes de l'utilisateur.

Elle peut être décomposée en plusieurs étapes :

#### 1<sup>ère</sup> étape :

Chargement en RAM du contenu de la ROM BIOS qui contient en particulier les pilotes ou gestionnaires des périphériques standards :

- Console,
- Imprimante
- Horloge
- Unités de lecteurs de disquettes et du disque dur.

La communication entre le DOS et ces pilotes se fait par l'intermédiaire de requêtes (demandes) envoyée par le DOS puis traduite par le BIOS en une série de commandes appropriées à chaque dispositif.

#### 2<sup>ème</sup> étape :

Recherche, dans la zone d'amorçage de l'unité de démarrage, du noyau du système d'exploitation puis son chargement en mémoire centrale.

Cette recherche se fait généralement par défaut dans la disquette placée dans le premier lecteur. Si cette unité ne contient pas de disquette, la recherche se poursuit automatiquement dans le disque dur.

En cas d'échec de cette recherche, c'est à dire si le noyau n'est pas trouvé, un message est affiché pour indiquer que le support utilisé (disquette ou disque dur) n'est pas un disque

système, c'est à dire qu'il ne contient pas de noyau, puis demander de le remplacer avant de pouvoir continuer.

**Remarque :** composition du noyau

Le noyau du système MS-DOS est principalement composé de trois fichiers ordinaires :

- IO.SYS
- MSDOS.SYS
- COMMAND.COM

**Définition :** On appelle fichier ordinaire tout ensemble d'informations stockées sur support magnétique (mémoire de masse).

Les opérations d'Entrée/Sortie relatives aux fichiers sont assurées par le système d'exploitation au niveau duquel chaque fichier se caractérise essentiellement par trois éléments :

- un nom (de 1 à 8 caractères valides)
- une extension (de 0 à 3 caractères)
- un ensemble d'attributs (A, R, H ou S).

Les attributs qui peuvent être activés ou désactivés par le système, servent à contrôler certaines opérations d'accès au fichier. Par exemple, un fichier ayant l'attribut R actif ne peut être utilisé qu'en lecture (R=ReadOnly). De ce fait, tant que l'attribut est actif le fichier ne peut être ni modifié ni supprimé.

**3<sup>ème</sup> étape :**

Recherche, dans une zone bien déterminée de l'unité de démarrage (racine ou root). D'un fichier particulier appelé CONFIG.SYS et destiné à regrouper des commandes de configuration.

Ces commandes, qui ne peuvent pas être directement lancées par l'utilisateur, permettent le choix de certains paramètres de configuration.

Ces commandes qui ne peuvent pas être directement lancées par l'utilisateur, permettent le choix de certains paramètres de configuration.

**Exemples :**

- Choix des conventions de notation relatives à la date, à l'heure, aux séparateurs décimaux, etc. (Command COUNTRY).
- Choix du nombre maximum de fichiers à ouvrir simultanément (command FILES).
- Déclaration d'un périphérique installable (mémoire haute, disque virtuel, etc.) et rechargement de son gestionnaire en RAM (command DEVICE).

**Remarques :**

- 1) En l'absence du fichier CONFIG.SYS, le système utilise des paramètres de configuration par défaut. Exemples :
  - Notations : Américaines,
  - Nombre maximum de fichiers à ouvrir en même temps : 8,



- 2) Le fichier CONFIG.SYS n'est consulté qu'une seule fois à chaque démarrage de la machine. Par conséquent, toutes modifications de ce fichier au cours d'une session de travail ne seront prises en considération que lors du prochain démarrage.

**4<sup>ème</sup> étape :**

Recherche de la racine de l'unité de démarrage (c'est à dire l'unité à partir de laquelle le noyau du système a été chargé en RAM) d'un second fichier particulier appelé AUTOEXEC.BAT.

Ce fichier sert à regrouper l'ensemble des commandes à exécuter automatiquement à chaque démarrage de la machine. Exemples :

Command	Utilité
KEYB	Permet le choix du type de clavier à utiliser.
PRPTOM	Fixe les informations qui doivent constituer le message d'attente ou l'indicatif du système.
PATH	Fixe les chemins à utiliser pour la recherche des fichiers exécutables.

En absence du fichier **AUTOEXEC.BAT**, il y a exécution automatique des deux commandes **DATE** et **TIME** qui permettent respectivement d'afficher et éventuellement d'initialiser la date et l'heure du système.

**5<sup>ème</sup> étape :**

Affichage de l'indicatif ou du message d'attente du DOS pour signaler que le système est prêt pour recevoir les commandes de l'utilisateur. Les informations constituant ce message dépendent des paramètres utilisés avec la commande **PROMPT** du fichier AUTOEXEC.BAT.

Ce message se présente le plus souvent sous l'une des deux formes suivantes :

- A:\>\_ (Si l'unité de démarrage est une disquette).
- C:\>\_ (Si le noyau est chargé à partir du disque dur).

En réponse à ce message, l'utilisateur peut introduire une ligne de commande en respectant la syntaxe générale de chaque commande à exécuter.

La ligne introduite sera lue et interprétée par l'interpréteur de commandes du système (qui fait partie du fichier COMMANDE.COM).

En cas d'erreur dans la ligne de commande, le système affiche un message d'erreur approprié. Sinon la commande sera exécutée puis l'indicatif du système à nouveau affiché.

**Exercice :**

Représenter à l'aide d'un pseudo-programme en langage algorithmique la procédure de démarrage décrite ci-dessus.

**IV. FORME GENERALE D'UNE LIGNE DE COMMANDE**

Comme pour tous les systèmes d'exploitation, les commandes du système MS-DOS correspondent en grande majorité à des **programmes paramétrés**.

Le lancement de ces commandes nécessite l'introduction de ligne de commandes qui se présentent selon la forme générale suivante :

< **Nom de la commande** > [liste de paramètres] [liste d'options] ↵

- Le nom d'une commande est formé soit par un verbe en Anglais (Copy, Format, Erase, Rename ,etc.) soit par un mnémonique (Cls, MD, CD, FC, etc.). Il doit être séparé du premier paramètre de la liste à l'aide d'un séparateur constitué par un ou plusieurs caractères d'espace .

- Les paramètres d'une commande servent à spécifier les éléments sur lesquels doit porter la commande.

**Exemples** : \* Pour renommer un fichier la commande **Rename** nécessite deux paramètres : l'ancien et le nouveau noms du fichier. Par contre, l'effacement d'un fichier nécessite qu'un seul paramètre : le nom du fichier à effacer .

Par ailleurs, certaines commandes ne nécessitent aucun paramètre (commandes non paramétrées).

\* la commandes Ver qui affiche le numéro de version utilisée ou la commande Cls qui permet d'effacer l'écran et de placer le curseur à sa première position.

- Les option éventuelles permettent d'agir sur la façon dont la commande sera exécutée.

**Exemple** : La ligne de commande **Sort<F**

Permet de classer les lignes d'informations issues du fichier texte F par ordre alphabétique croissant.

Si on lui ajoute l'option /r , le classement sera effectué par ordre décroissant.

- La fin de la ligne de commande doit être signalée à l'aide de la touche de validation ↵ (touche Entrée ou Enter ou Return).

- L'édition ou la correction d'une ligne de commande peut être effectuée avant sa validation grâce aux touches de déplacement du curseur (←, →, ↑, ↓, End, Home) ainsi que les touches d'insertion (Ins) et de suppression (Del ou Suppr).

# CHAP 3 : GESTION DES SUPPORTS MAGNETIQUES

## I. NOTIONS PRELIMINAIRES

---

### 1. Unité Logique :

C'est un nom spécial grâce auquel le système peut identifier un lecteur donné.

- Chaque lecteur correspond à un dispositif physique bien déterminé : Disquette, Disque dur, CD-ROM,...
- Par convention les noms associés à ces lecteurs sont constitués d'une seule lettre suivie du caractère " :".
- Les unités logiques les plus couramment utilisées sont :

Unité	Dispositif correspondant
A :	Premier lecteur de disquettes,
B :	Deuxième lecteur de disquettes,
C :	La totalité ou une partie du disque dur,
D :	Une partition du disque dur, ou lecteur CDROM

### 2. Partitionnement d'un disque dur

Lorsque la capacité de stockage d'un disque dur est suffisamment élevée, il peut s'avérer commode de le subdiviser en plusieurs partitions. Dans ce cas, chaque partition sera considérée comme une unité logique à part et peut être désignée à l'aide des noms D:, E: ou F: .

L'opération de partitionnement est assurée par l'utilitaire FDISK qui propose un menu permettant le choix d'autres opérations à réaliser telles que l'activation ou la suppression d'une partition existante.

Intérêts d'un partitionnement : Meilleure organisation des informations sur le disque, Possibilité d'utiliser plus d'un système d'exploitation sur une même machine. Exemple :

- Partition C : ---> MS-DOS,
- Partition D : ----> UNIX

Si le disque est de capacité supérieur à 2 Go, Il sera obligatoire de le partitionner si vous voulez y installer MSDOS ou Windows 95 , ces deux systèmes ne gèrent pas des lecteur de capacité supérieur à 2 Go.

### 3. Unité de démarrage ou de lancement

C'est l'unité à partir de laquelle le noyau du système d'exploitation est chargé en mémoire vive de l'ordinateur au cours de la procédure de démarrage.

### 4. Interpréteur de commandes

C'est le programme qui permet la lecture et l'analyse syntaxique des lignes de commandes fournies par l'utilisateur, ainsi que le chargement éventuel en RAM des fichiers contenant

les codes compilés des commandes à exécuter. Il s'agit en général du fichier COMMAND.COM qui comporte une partie résidente et une partie transitoire. Ce fichier peut être remplacé par n'importe quel autre programme capable de réaliser la même tâche (par exemple un interpréteur développé par l'utilisateur). Dans ce cas, le programme en question doit être spécifié dans le fichier de configuration CONFIG.SYS grâce à la commande SHELL.

### 5. Notions de commande externe et de commande interne

Une commande externe est un programme exécutable dont le code compilé est stocké dans un fichier à part (avec l'extension .EXE ou .COM),

Les commandes internes sont toutes regroupées au sein d'un même fichier (le fichier COMMAND.COM). Elles sont résidentes en RAM. Par conséquent, leur exécution est plus rapide que celle d'une commande externe.

### 6. Activation d'une unité

Elle se fait tout simplement en tapant le nom de l'unité à activer comme réponse au message d'attente du DOS puis de valider la ligne tapée. Le nom de l'unité active est toujours rappelé dans l'indicatif du DOS. Il est utilisé par défaut pour toutes les opérations de lecture et d'écriture relatives aux disques lorsque aucune autre unité n'est spécifiée dans la ligne de commande qui nécessite une telle opération. Exemple : La ligne de commande : **COPY A :F** Lancée à partir de l'indicatif : **C:/>\_** Permet de faire une copie du fichier F de l'unité A vers l'unité active **C:** car la commande ne précise pas la destination de la copie. Par ailleurs, la ligne de commande :

**COPY F A :** Réalise une copie sur l'unité de destination **A:** d'un fichier **F** qui sera recherché par défaut dans l'unité active **C:** car la commande ne précise pas la source de la copie.

Enfin, si on tape la ligne : **A :** L'indicatif devient : **A:\>\_**

### 7. Notion de disque virtuel

C'est une zone de la mémoire vive (RAM) qui peut être réservée puis considérée comme un disque à accès rapide auquel on peut associer les noms logiques D :, E :, F :,...

Les disques virtuels sont considérés comme des périphériques installables. Leur pilote (ou gestionnaire ou driver) se trouve dans le fichier RAMDRIVE.SYS ou VDISK.SYS.

La création d'un disque virtuel se fait au moment du démarrage de la machine grâce à la commande de configuration DEVICE.

Le nombre maximum de disques virtuels qui peuvent être créés n'est limité que par la taille mémoire disponible.

La commande de configuration LASTDRIVE permet de fixer le dernier nom logique qui peut être accepté par le système. Exemple : LASTDRIVE = J Ne permet l'utilisation que des dix premiers noms A :, B :, ...

Une fois créé, un disque virtuel peut être utilisé de la même façon que les unités de disque réelles A, B et C. Il permet ainsi de réaliser un gain de temps en évitant des accès répétitifs aux disques.

## II. PRINCIPALES COMMANDES RELATIVES AUX DISQUES

### 1. Formatage ou initialisation d'une disquette

#### A. Structure physique d'une disquette

Physiquement l'espace de stockage des informations sur une disquette est subdivisé en plusieurs pistes circulaires et concentriques.

- Chaque piste est divisée à son tour en plusieurs secteurs.
- Chaque secteur permet de stocker 512 Octets par face.
- Le tableau suivant donne les principales caractéristiques de quelques disquettes utilisées :

Diamètre (en pouces)	Densité	Nombre de pistes	Nombre de secteurs par piste	Capacité de stockage
5" ¼	DD	40	9	360 Ko
	HD	80	15	1,2 Mo
3" ½	DD	80	9	720 Ko
	HD	80	18	1,44 Mo
	ED	80	36	2.88 Mo

DD= Double Densité; HD = Haute Densité ED = extra densité

#### B. Formatage physique

C'est l'opération qui consiste à tracer physiquement les pistes et les secteurs sur le support magnétique. Cette opération n'est pas assurée par le système d'exploitation. Elle se fait en principe par le constructeur au moment de la fabrication du support.

#### C. Formatage logique

Cette opération est indispensable afin que le système puisse lire ou écrire des informations sur le support. Elle est assurée par un utilitaire du système (programme FORMAT) et consiste en :

- L'identification de tous les secteurs utilisables (non défectueux),
- La création du répertoire racine du support (voir chapitre suivant),
- La création de la table d'allocation des fichiers (FAT),

#### D. Définition de la FAT (File Allocation Table)

C'est un fichier particulier qui contient des informations sur l'état de chaque secteur du disque qui peut être :

libre, alloué à un fichier donné, réservé au système, non utilisable ou défectueux.

La consultation de cette table est indispensable pour que le système puisse accéder aux différents secteurs alloués à chaque fichier du disque.

Vu cette importance de la FAT, elle est créée en double exemplaire. Ceci permet de réduire le risque de perte des informations stockées sur le support si l'un des deux exemplaires est endommagé.

#### E. Syntaxe de la commande FORMAT

**Format <Nom de Lecteur> [Liste d'options]**

Principales options :


- |       |   |
|-------|---|
| /S    | Copie du noyau après formatage,   |
| /F :N | Spécifie la capacité de stockage N de la disquette en Ko<br>(180, 360, 720, |

**Exemple :** **FORMAT A: /F :720**

Cette commande permet de formater une disquette de 720 Ko placée dans le lecteur A.

- Remarque : Les disquettes de type haute densité (HD) ne peuvent pas être formatées à l'aide d'un lecteur double densité (DD). La réciproque n'est pas vraie.

## 2. La commande LABEL

- But : Attribuer un nom de volume à un disque
- Syntaxe : **LABEL [unité :] [nom de volume]**
- Remarques
  - un nom de volume peut comporter jusqu'à 11 caractères espacés ou non,
  - si aucun nom n'est spécifié, un message le demande,
  - si la touche  est tapée en réponse à ce message, alors l'ancien nom sera affecté.
  - Les commandes DIR et VOL affichent aussi les noms de volumes,

## 3. La commande VOL

- But : Affiche le nom de volume (LABEL) et du numéro de série d'un disque.

## 4. La commande DiskCopy

- But : Dupliquer une disquette
- Syntaxe : **DiskCopy [U1 :] [U2 :] [/1]**
  - U1 = Unité source,
  - U2 = Unité destination,
  - /1 permet de ne copier que la première face de la disquette source.
- Remarques :
  - les deux disquettes source et destination doivent avoir le même format
  - La commande ne s'applique pas au disque dur
  - Il n'est pas nécessaire que la disquette cible soit formatée. S'elle contient des informations, elles seront perdues.
  - La copie se fait secteur par secteur.
  - Si aucune unité n'est spécifiée la commande demande l'introduction des deux disquettes source et cible.

## 5. la commande DiskComp

- But : Vérifier ces deux disquettes de même format sont identiques ou non.
- Syntaxe : **DiskComp [U1 :] [U2 :]**
- Remarques :
  - Si aucune unité n'est spécifiée, la commande demande l'introduction successive des deux disquettes à comparer.

- La comparaison se fait secteur par secteur. Par conséquent, deux disquettes qui contiennent les mêmes fichiers ne sont pas nécessairement identiques.

#### 6. La commande ChkDsk

- But : Détection et correction d'erreurs éventuelles de fragmentation sur disque .
- Syntaxe : **ChkDsk [U :] [chemin] [fichier (s)] [/F] [/V]**
- Options :
  - /F permet la correction des erreurs rencontrées
  - /V affiche la liste des fichiers au cours de la vérification
- Remarques :
  - Si aucun fichier ni répertoire n'est spécifié, la commande s'applique à la totalité de disque,
  - Dans tous les cas la commande affiche un relevé sur l'état de disque. Ce relevé contient plusieurs information utiles tel que : Taille totale d'espace disque, Nombre de répertoire, Nombre de fichiers, Nombre d'octet libre sur le disque ...etc.

#### 7. la commande SUBST

- But : Affecter un nom de lecteur logique à un répertoire
- Syntaxe : **SUBST [Unité :] [Chemin]**
- Remarques :
  - La commande revient en réalité à représenter un chemin d'accès à l'aide d'un nom d'unité virtuelle,
  - Une fois créée, une unité virtuelle peut être activée et utilisé de la même façon que les unités réelles A :, C :
  - Le nombre de maximum d'unité possible dépend de la commande de configuration LASTDRIVE (la valeur par défaut est égale au nom qui suit immédiatement le dernier lecteur réel utilisé),
  - Le système réserve une structure de données en mémoire pour toutes les unités spécifiées à l'aide de la commande LASTDRIVE,
  - Utilisée sans paramètre la commande permet d'afficher les affectations actives,
- Exemple : La commande : **Subst E: C:\PASCAL\BIN**  
Affecte le nom logique E: au sous répertoire BIN du répertoire PASCAL de l'unité C:. une fois cette commande est exécutée, l'unité virtuelle E: peut être activée et utilisée de la même façon que les unités A :, B : et C :.

# CHAP 4 : GESTION DES REPERTOIRES

## I. DEFINITIONS ET GENERALITES

### 1. Définition d'un répertoire

Un répertoire est un dossier (ou catalogue) destiné à regrouper des fichiers et des sous-répertoires pouvant être utilisés indépendamment des autres dossiers d'un même disque.

### 2. Utilité des répertoires

Eviter de regrouper ensemble des informations de natures différentes (utilitaires du système, compilateurs, logiciels d'applications, etc.).

Permettre le partage d'un même support entre plusieurs applications ou plusieurs utilisateurs.

Eviter des suppressions accidentelles d'informations. En effet, une commande telle que **DEL \*.\*** permet d'effacer tout le contenu du répertoire de travail. Si tout le contenu du disque est rassemblé au sein de cet unique répertoire, alors il sera entièrement effacé.

### 3. Organisation logique d'un disque

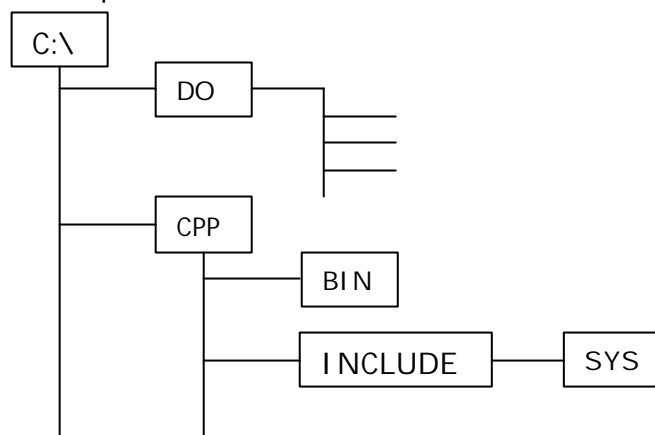
En plus de la possibilité de rassembler des fichiers de même famille, un répertoire peut contenir également des sous-répertoires. Chaque sous-répertoire pouvant être à son tour subdivisé en plusieurs autres sous-répertoires et ainsi de suite.

Ceci conduit à une structure logique du disque qui peut être représentée sous la forme d'un arbre « structure arborescente ».

La racine de l'arbre est représentée par le répertoire principal qui se crée automatiquement au moment du formatage du disque et qui est désigné par le symbole spécial « \ »,

Les branches correspondent aux sous-répertoires et les feuilles aux fichiers.

Exemple :





#### 4. Répertoires particuliers

##### A. Répertoire actif

C'est le répertoire qu'utilise le système par défaut pour lire ou écrire des informations sur un disque lorsque aucun autre répertoire n'est spécifié dans une ligne de commande.

- Lorsqu'un disque est utilisé pour la première fois au cours d'une session de travail, le système active automatiquement son répertoire racine.
- La commande **Cd** permet à chaque instant d'activer un répertoire pour chaque unité de disque.
- Le répertoire actif de l'unité de travail est généralement rappelé dans le message d'invite du système
- Le répertoire de travail est désigné par le symbole spécial " . ".

##### B. Répertoire parent

C'est le répertoire qui se trouve à un seul niveau au-dessus du répertoire de travail, et ce sans quitter ce dernier.

#### 5. Chemin d'accès

##### A. forme générale

[\] [Rép1\] [Rép2\] [Rép3\] [...]

Les différents sous-répertoires constituant la chaîne doivent être séparés par le caractère " \ ",

Un chemin est dit absolu s'il commence à partir du répertoire racine, c'est à dire si son nom commence par le caractère " \ ".

Un chemin est dit relatif s'il commence à partir du répertoire courant.

##### B. Remarque

Un nom de fichier peut toujours être précédé par le Nom de l'unité qui le contient suivi du chemin qui localise le fichier sur cette unité.

Exemple : Pour afficher le contenu d'un fichier nommé **F.DOC** situé dans le sous-répertoire **CPP** de l'unité **C** : quel que soit le répertoire courant on peut utiliser la ligne de commande :

**TYPE C:\CPP\F.DOC** ↵

Dans ce cas, le fichier ne sera pas recherché dans le répertoire de travail de l'unité active mais selon le chemin spécifié dans l'unité **C** :

## II. PRINCIPALES COMMANDES RELATIVES AUX REPERTOIRES

---

### 1. Création d'un nouveau répertoire

MD ou Mkdir

Syntaxe générale : MD [U :] [Ch] <Nom de Rép>

**Exemples :** a) La ligne de commande : C:\DOS> MD com ↵

Crée un sous répertoire nommé com dans le sous-répertoire DOS de la racine de l'unité C:

b) MD \TC ↵ crée un sous-répertoire TC dans le répertoire racine de l'unité active.

c) MD A:R1 ↵ crée dans le répertoire actif de l'unité A un sous-répertoire de nom R1.

### 2. Activation d'un répertoire CD ou ChDir

Syntaxe générale : CD [U :] [Ch] <Nom de Rép>

**Exemples :**

a) CD\ ↵ ---> Activation du répertoire racine de l'unité en cours.

a) C:\> CD A:\R1 ↵ Entraîne l'activation du sous-répertoire R1 de la racine de l'unité A sans changer l'unité courante C, c'est à dire sans activer l'unité A..

### 3. Suppression d'un répertoire vide RD ou Rmdir

Syntaxe générale : RD [U :][Ch] <Nom de Rép>↵

**Exemple :** RD A:\R1 ↵

### 4. Affichage d'une arborescence Tree

Syntaxe générale : Tree [U:] [Ch] [/F] [/A]↵

**/F** Affiche aussi les noms des fichiers contenus dans les différents répertoires de l'arborescence,

**/A** Indique à la commande de n'utiliser que des caractères ASCII pour représenter les lignes de l'arborescence,

**Exemples:** a- Tree ↵ Affichage de l'arborescence de l'unité active à partir du répertoire courant et sans indication de fichiers contenus dans chaque sous-répertoire de cette arborescence,

b- Tree C:\TC /F ↵ Affichage de l'arborescence qui commence à partir du répertoire TC de la racine de l'unité C avec indication de chaque nom de fichier rencontré.

### 5. Spécification de plusieurs chemins d'accès aux fichiers exécutables

Path

**Rappel :** Un fichier exécutable est un fichier qui contient le code compilé d'un programme (extension .EXE ou .COM) ou une liste de commandes à exécuter en lot (fichier bath avec l'extension .BAT).

Par défaut, la recherche d'un tel fichier se fait dans le répertoire courant de l'unité active. Si le fichier n'est pas trouvé et si aucun chemin de recherche n'est déjà spécifié par la commande PATH, alors le système affiche le message : *Nom de commande ou de fichier incorrect*

Syntaxe de la commande PATH : **PATH [U:] [Ch] [<;> [U:] [Ch]...**

#### **6. Fonctionnement :**

Une fois une ligne de commande Path est exécutée, la recherche des fichiers exécutables ne sera plus limitée au répertoire courant, mais se poursuivra, si nécessaire, aux différents répertoires spécifiés dans cette ligne de commande.

#### **7. Intérêt de la commande :**

Eviter d'avoir à activer un répertoire avant de lancer l'exécution d'un fichier qu'il contient, ou d'avoir à spécifier le chemin d'un fichier exécutable qui n'appartient pas au répertoire courant.

Exemples : a) PATH C:\;C:\PASCAL\BIN;C:\TC\BIN ←↵

Dans ce cas, la recherche se fera d'abord dans le répertoire actif de l'unité courante puis se poursuivra si nécessaire suivant les quatre répertoires spécifiés.

b) PATH ←↵ -- > Affichage de la commande Path active. C'est à dire des différents chemins utilisés en ce moment pour la recherche des fichiers exécutables.

c) PATH ; ←↵ --- > Désactivation de tous les chemins éventuellement pré activés pour la recherche. Après cette commande la recherche sera limitée uniquement au répertoire de travail.

# CHAP 5 : OPERATIONS SUR LES FICHIERS

## I. NOTIONS COMPLEMENTAIRES

### 1. Les caractères génériques

#### A. Définition

Un caractère générique (ou caractère de remplacement ou "joker") est un symbole qui sert à remplacer un ou plusieurs autres caractères dans le nom ou dans l'extension d'un fichier.

#### B. Intérêt des caractères génériques

C'est de pouvoir désigner plusieurs fichiers ayant une propriété commune à l'aide d'un même nom dit "nom générique". Ceci peut être particulièrement utile lorsqu'on veut réaliser une même opération sur un grand nombre de fichiers. Par exemple : faire une copie de sauvegarde de tous les programmes sources écrits en langage C. Dans ce cas, la commande de copie peut être lancée à l'aide d'une seule ligne de commande à condition de lui faire passer comme premier paramètre le nom générique représentant l'ensemble des fichiers à copier.

#### C. Caractères génériques reconnus par le DOS

Les caractères génériques peuvent changer d'un système d'exploitation à un autre. Ceux utilisés par le système DOS sont : -l'astérisque (\*) qui permet de remplacer toute chaîne constituée de 0 à 8 caractères,

-le point d'interrogation (?) qui sert à remplacer un seul caractère (y compris le caractère vide).

Exemples de noms génériques :

Nom générique	Signification
*.C	Tous les fichiers ayant l'extension ".C",
F?.*	Fichiers dont le nom comporte au plus deux caractères et commence par la lettre F,
A*??	Fichiers dont le nom commence par la lettre A et dont l'extension comporte au plus deux caractères,
*.	Fichiers dont l'extension est vide.

### 2. Les fichiers spéciaux

Un fichier spécial (ou fichier de périphérique) est un moyen utilisé par le système d'exploitation pour permettre aux utilisateurs d'utiliser certains périphériques (notamment l'écran, le clavier et l'imprimante) de la même manière que les fichiers normaux sur disque.

Pour cela, chaque périphérique est représenté symboliquement par un nom de fichier prédéfini auquel l'utilisateur peut faire référence pour accéder au dispositif physique

correspondant. Le transfert d'informations entre ce dispositif est la mémoire centrale étant assuré par le système d'exploitation de façon transparente pour l'utilisateur. Les principaux fichiers spéciaux utilisés par le DOS sont :

Fichier spécial	Dispositif correspondant
CON	Console (clavier+Ecran),
PRN (ou LPT1)	Imprimante connectée au premier port //,
LPTi (i=1,2,3,4)	Imprimante connectée au ième port //,
COM (ou AUX)	Imprimante liée au premier port série,
COMi (i=1,2,3,4)	Imprimante liée au ième port série,
NUL	Aucun dispositif.

Remarques :

- Les noms de fichiers spéciaux sont des noms réservés, c'est à dire qu'ils ne peuvent pas être données à des fichiers ordinaires sur disque.
- Le fichier spécial nommé NUL ne représente aucun périphérique réel. Il peut être utile lorsque des informations doivent nécessairement être envoyées vers un fichier de sortie alors qu'aucun fichier ne doit, ou ne peut, être créé. Exemple : Certains compilateurs (comme le FORTRAN) exigent un nom de fichier pour la sortie des messages d'erreurs, pour ne pas garder ces messages, il suffit de fournir le nom NUL comme fichier de sortie des erreurs.

**3. Les redirections d'entrée/Sortie**

**A. Introduction**

Par défaut, les résultats fournis par les commandes du DOS sont affichés sur l'écran : on dit que l'écran est le fichier de sortie standard.

De même, lorsqu'une commande nécessite la spécification du fichier contenant les données à traiter alors qu'aucun fichier n'est précisé dans la ligne de commande, ces données peuvent être automatiquement lues à partir du clavier : on dit que le clavier est le fichiers d'entrée standard ou par défaut.

De façon plus générale, les données d'une commande peuvent être lues à partir d'un fichier quelconque et ses résultats envoyés vers n'importe quel fichier. Pou cela, il suffit d'effectuer une redirection d'entrée et/ou de sortie de la commande.

Ces redirections peuvent être effectuées grâce à des symboles spéciaux ayant chacun une signification spéciale pour le compilateur.

**B. Symboles de redirection**

Nom générique	Signification
<	Redirige l'entrée vers un fichier autre que le clavier,
>	Redirige la sortie vers un fichier autre que l'écran
>>	Ajoute la sortie d'une commande à la fin d'un fichier existant (ou à créer).

Exemples :

`DIR A:>PRN` ← Impression du contenu du répertoire actif de la disquette placée dans le lecteur A;

`DIR C:\DOS> F` ← Création, dans le répertoire actif de l'unité courante, d'un fichier nommé F et stockage dans ce fichier du contenu du répertoire C:\DOS. Si un fichier de même nom existe déjà il sera perdu.

`TYPE F>>G` ← Ajout du contenu de F à la fin du fichier G. Si G n'existe pas, il sera créé (Dans ce cas, la commande est équivalente à `type F>G` )

`SORT <F>G` ← Lecture des lignes d'informations stockées dans F, puis leur classement par ordre alphabétique et écriture du résultat dans un nouveau fichier G.

`SORT>PRN` ← Lecture d'informations à partir du clavier, à raison d'une ligne à la fois, puis classement de ces lignes et impression du résultat. La fin de la saisie aura lieu lorsqu'une ligne ne contenant que le caractère spécial EOF (CTRL+Z ou F6) sera validée.

#### 4. Enchaînement de commandes

a) But de l'opération : C'est de pouvoir effectuer plusieurs traitements successifs sur un même ensemble de données grâce à une seule ligne de commandes.

b) Principe : Il consiste tout simplement à rediriger la sortie de chaque commande vers l'entrée de la commande suivante. Pour cela, chaque couple de commandes successives doivent être séparées par le caractère spécial d'enchaînement "|" (de code ASCII =124).

c) Exemples :

`TYPE F.C | MORE` ← Affiche, Une page écran à la fois, du contenu du fichier F.C

`DIR | SORT >PRN` ← Classement par nom de tous les fichiers du répertoire actif puis impression du résultat.

#### 5. Notion de filtre

##### A. Définition :

Un filtre est une commande dont le rôle est de tester les lignes du fichier texte qu'elle reçoit en entrée pour n'en restituer en sortie que celles qui obéissent à une certaine condition. Les commandes de filtre du système DOS sont : FIND, SORT et MORE.

##### B. la commande FIND

**But** : c'est de rechercher une chaîne de caractères au sein d'un fichier texte.

Syntaxe générale d'utilisation : **FIND <"Ch"> [Fichier] [/V] [/C] [/N]**

Le paramètre obligatoire "**Ch**" est la chaîne à rechercher. Elle doit être mise entre deux guillemets et sera recherchée dans le fichier texte dont le nom peut être fourni comme second paramètre de la commande. Si ce nom est omis, la recherche sera effectuée dans le fichier d'entrée standard (c'est à dire le clavier).

**options** : /V Ne fournit en sortie que les lignes qui ne contiennent pas la chaîne à rechercher (inverse le filtre).

/C Ne fournit que le nombre de lignes trouvées (c'est à dire qui contiennent ch).

/N Fait précéder chaque ligne sortie par son numéro dans le fichier.

Exemples :

DIR | FIND ".BAK" /V Liste de tous les fichiers du répertoire courant sauf ceux ayant l'extension ".BAK",

FIND "readln" P1.pas /N affiche toutes les lignes qui contiennent le mot readln en faisant précéder chaque ligne par son numéro dans le fichier.

DIR C:\TC\BIN | FIND "4.99" Sortie de tous les fichiers du répertoire C:\TC\BIN qui ont été créés ou modifiés durant le mois d'Avril 1999.

### C. La commande SORT:

<Commande1> | SORT [/R] [/+N]

SORT <Fichier [/R] [/+N]

Dans le premier cas, les lignes à classer constituent le fichier de sortie d'une première commande à enchaîner avec SORT. Dans le second cas ces lignes seront lues à partir du fichier d'entrée spécifié après le symbole "<".

**options** : /R Tri dans l'ordre décroissant au lieu de l'ordre par défaut (croissant).

/+N le tri se fait à partir du Nième caractère de chaque ligne au lieu du premier (par défaut N=1).

Exemples :

DIR | SORT /R /+14 Affichage de la liste de tous les fichiers du répertoire courant, classés par ordre décroissant selon le champ taille,

SORT <F> LPT1 Impression des lignes issues du fichiers F, après les avoir classées par ordre alphabétique croissant.

### D. La commande MORE

*But* : Provoquer un arrêt d'affichage provisoire à chaque fois que l'écran est plein. La reprise de l'affichage peut être effectuée en appuyant sur une touche quelconque.

C'est une commande très utile pour la visualisation de fichiers contenant plusieurs lignes d'informations. Elle peut être utilisée selon deux syntaxes différentes :

**MORE** <Fichier ↵

**Commande** | **MORE** ↵

Exemple : Les deux lignes de commandes suivantes sont équivalentes : MORE<P.PAS ↵  
2) TYPE P.PAS | MORE ↵

Elle permettent toutes les deux d'afficher, écran par écran, le contenu du fichier source P.PAS.

## II. ETUDES DES PRINCIPALES COMMANDES RELATIVES AUX FICHIERS

---

### 1. Manipulation des attributs d'un fichier

Syntaxe : **ATTRIB** [Unité:] [Chemin] [Fichier(s)] [+X|-X] [/S]

X désigne l'un des attributs A, R, H ou S,

+X positionne (ou active) attribut X,

-X désactive l'attribut X,

Le signe " | " désigne l'opérateur "OU EXCLUSIF",

\* Les noms des fichiers à traiter peuvent, comme toujours, être précédés de leur chemin d'accès ainsi que du nom de l'unité qui les contient.

\* si aucun fichier n'est donnée comme paramètre, la commande s'applique par défaut à tous les fichiers du répertoire de travail.

\* Si on spécifie un chemin d'accès sans préciser de fichiers, la commande s'applique au dernier répertoire de ce chemin (qui n'est qu'un fichier particulier).

Enfin l'option/S permet d'appliquer la commande également aux fichiers des sous-répertoires du chemin spécifié.

-Exemples : 1) **ATTRIB +r C:\DOS\\*.COM** ↵

Activation de l'attribut lecture seulement (+r) pour tous les fichiers du sous-répertoire DOS de l'unité C, qui ont l'extension .COM,

2) **ATTRIB/S** ↵ Affichage des attributs actifs de tous les fichiers du répertoire courant (y compris ceux appartenant à ses sous-répertoires).

## 2. Lecture d'un fichier texte

**TYPE [Unité:] [Chemin]<Fichier>**

Exemples :

**TYPE C:\AUTOEXEC.BAT** ↵

**TYPE A:P1.C | MORE** ↵

**TYPE C:\CONFIG.SYS> PRN** ↵

Remarque :

L'affichage peut être momentanément interrompu grâce à la combinaison de touches CTRL+S puis repris à l'aide de n'importe quelle touche.

## 3. Duplication de fichiers : la commande COPY

Syntaxe simplifiée : **Copy [U1][Ch1]<Source(s)> [U2:][Ch2][Cible(s)] [/V]**

Caractéristiques :

\* La commande ne s'applique pas aux fichiers ayant les attributs S ou H,

\* Quels que soient les attributs des fichiers sources, les fichiers cibles auront l'attribut par défaut A (car ce sont des fichiers à créer).

\* Si le second paramètre (cible est omis, les fichiers résultant de la copie auront les mêmes noms que les fichiers sources correspondants.

L'option /V sert à activer les vérifications automatiques des écritures sur le disque. Ceci permet de s'assurer que les copies sont effectuées correctement.

Exemples : 1) **COPY CON A:AUTOEXEC.BAT** ↵

Création dans le répertoire actif de l'unité A, d'un fichier nommé AUTOEXEC.BAT puis écriture dans ce fichier du texte saisi au clavier jusqu'à l'introduction d'une ligne ne contenant que le caractère spécial "fin de fichier" (CTRL+Z ou F6).

2) **COPY C:\TC\BIN\\*.C A:\*.BAK** ↵

Copie sur l'unité A de tous les programmes sources en C, issus du répertoire \TC\BIN de l'unité C. Les fichiers ainsi copiés gardent le même nom mais auront ".BAK" comme extension.



Remarque : La commande COPY sert aussi à concaténer des fichiers. Pour cela, on peut utiliser la syntaxe simplifiée suivante : **COPY <Nom(s)1>+<Nom(s)2> + .... <F>**

Où F est le nom de fichier résultant de la concaténation de tous les autres fichiers spécifié sur la ligne de commande.

#### 4. Copie de fichiers et de sous-répertoires

En plus de la possibilité de dupliquer des disquettes (DISKCOPY) ou des fichiers (COPY), le système DOS offre aussi la possibilité de copier toute une partie d'une arborescence. C'est à dire tout le contenu d'un répertoire, y compris ses sous-répertoires et les éventuels qu'ils contiennent.

La commande à utiliser pour cela est : XCOPY.

Syntaxe générale simplifiée : **XCOPY [U1][Ch1]<Nom(s)2> [Options]**

Principales options :

/S Copie aussi les sous-répertoires non vides,

/S/E Copie tous les sous-répertoires même s'ils sont vides. ( /E(Empty) ne peut pas être utilisée sans l'option S (Sub)).

Remarque : Si la nature du paramètre cible (fichier ou répertoire) n'est pas précisée (le caractère "\" à la fin du nom signifie qu'il s'agit d'un répertoire), alors la commande demande s'il s'agit d'un fichier ou d'un répertoire à travers un message du type : "Nom2 représente -t-il un nom de fichier ou de répertoire sur le lecteur cible"

(F=Fichier, R=Répertoire)?\_

Si la réponse est R, alors il y aura création du répertoire cible puis copie du paramètre source dans ce répertoire. Dans le cas contraire, il y aura création d'un fichier cible unique puis écriture dans ce fichier de tout le contenu de la source (si la source est un répertoire, tous ses fichiers seront concaténés dans le fichier cible créé).

Exemples : 1) **XCOPY C:\DOS A:\DOS** ↵ On reçoit le message:

"DOS représente -t-il un nom de fichier ou de répertoire dans le lecteur cible (F=Fichier, R=Répertoire)?\_"

1<sup>er</sup> Cas : Réponse = F ⇒ Création d'un fichier unique sur A appelé DOS puis concaténation dans ce fichier de tout le contenu du répertoire C:\DOS?

2<sup>ème</sup> Cas : Réponse = R ⇒ Copie dans le répertoire DOS de l'unité A du contenu du répertoire C:\DOS. Si l'unité A ne contient pas de répertoire appelé DOS, alors il sera créé.

2) **XCOPY C:\DOS A:\DOS\ /P/M/S/E** ↵

Dans ce cas, il n'y a pas d'ambiguïté car le paramètre cible est un répertoire. Donc la commande va effectuer les tâches suivantes :

- Création du répertoire DOS dans l'unité A (Si ce répertoire n'existe pas déjà),
- Copie dans ce répertoire ainsi créé de tous les fichiers sources avec demande de confirmation pour chaque fichier à copier (/P) et marquage des fichiers copiés (/M).

3) **XCOPY C:\\*.BAT BAT\** ↵

Copie de tous les fichiers de la racine de l'unité C ayant l'extension .BAT dans un sous-répertoire appelé BAT, à créer dans le répertoire de travail courant.

### 5. Changement de noms de fichiers : La commande REN

Syntaxe : **REN [U1][Ch] <Nom(s)1> <Noms(s)2>**

Exemples : 1) **REN \*.TXT \*.DOC** ↵ transforme toutes les extensions .TXT en .DOC.

1) **REN ??.\*.DAT** Ajoute l'extension .DAT à tous les fichiers du répertoire courant dont le nom comporte au maximum deux caractères et qui n'ont pas d'extension.

