

# SYSTÈME D'EXPLOITATION MS DOS



# SOMMAIRE

## Que se passe-t-il quand on allume l'ordinateur ?

- Autotest
- Le BIOS et ses tâches
- Chargement du système d'exploitation
- Qu'est ce qu'un système d'exploitation
- Les fichiers CONFIG.SYS et AUTOEXEC.BAT

## CONFIGURATION MEMOIRE

- Base de gestion mémoire
- Les différents types de mémoires
  - \* Mémoire paginée EMS
  - \* Mémoire étendue XMS
  - \* Mémoire supérieure UMB
  - \* Mémoire haute HMA

## FICHIERS DE CONFIGURATION

- Gestionnaires de périphériques
- Gestionnaires Externe et Interne

## FICHER CONFIG.SYS

- Gestionnaires de périphérique
  - \* ANSY.SYS
  - \* DBLSPACE.SYS
  - \* EGA.SYS
  - \* HIMEM.SYS
  - \* POWER.EXE
  - \* SETVER.EXE
  - \* COUNTRY.SYS
  - \* DISPLAY.SYS
  - \* EMM386.EXE
  - \* INTERLNK.EXE
  - \* RAMDRIVE.SYS
  - \* DRIVERS.SYS
- Commande du CONFIG.SYS
  - \* BREAK
  - \* DOS
  - \* FILES
  - \* INSTALLHIGH
  - \* REM
  - \* STACKS
  - \* BUFFERS
  - \* DRIVPARM
  - \* INSTALL
  - \* LASTDRIVE
  - \* SHELL
  - \* SWITCHES
  - \* COUNTRY
  - \* FCBS
  - \* NUMLOCK

## FICHER AUTOEXEC.BAT

- Programme résident TSR
  - \* APPEND.EXE
  - \* GRAPHICS.COM
  - \* MODE.COM
  - \* PRINT.EXE
  - \* SMARTDRV.EXE
  - \* DOSKEY.COM
  - \* INTERSVR.EXE
  - \* MOUSE.COM
  - \* SHARE.EXE
  - \* UNDELETE.EXE
  - \* FASTOPEN.EXE
  - \* KEYB.COM
  - \* MSCDEX.EXE
  - \* VSAFE.COM
- Mémoire d'environnement
  - \* BREAK
  - \* DIRCMD
  - \* MSDOSDATA
  - \* TEMP
  - \* TMP
  - \* COMPSPEC
  - \* PATH
  - \* WINPMT
  - \* CONFIG
  - \* PROMPT
- Ignorer CONFIG.SYS et AUTOEXEC.BAT au démarrage

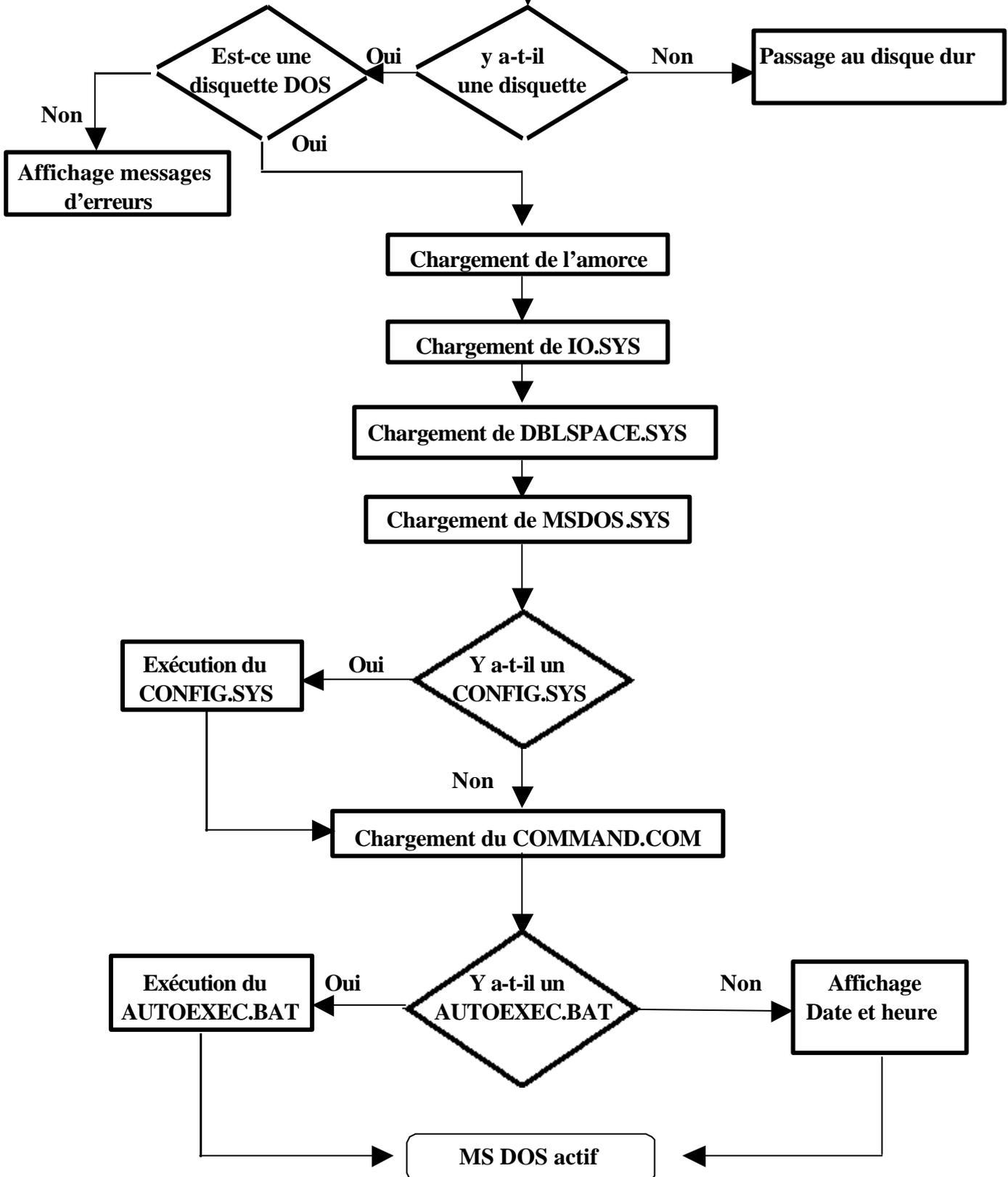
## REALISATION DE CONFIG.SYS

## REALISATION DE AUTOEXEC.BAT

## OPTIMISER LA CONFIGURATION AVEC MEMMAKER.EXE

Mise sous tension

Initialisation BIOS : Autotest et configuration ( SETUP )



## Que se passe-t-il quand on allume l'ordinateur ?

Le fait de mettre l'ordinateur en marche signifie deux choses : D'une part, il faut que tous les composants reçoivent du courant et qu'il soient initialisés, c'est à dire ramenés à un état initial toujours identique à lui même.

Exemple : La même position de départ pour les têtes de lecture des lecteurs de disquettes.

D'autre part, il faut que le système d'exploitation du PC soit chargé en mémoire principale, car il doit pouvoir contrôler les logiciels que vous utilisez et recevoir leurs commandes afin de les traiter. On étudiera le cas du MS DOS.

Le démarrage du système comprend donc un double processus :

- La mise en route du PC lui-même ( Hardware )
- Le chargement du système d'exploitation ( Soft )

Le processus de démarrage est désigné par le terme anglo - Américain suivant : **BOOTER**

Une fois que l'alimentation atteint la tension optimale et que tous les composants sont alimentés en courant électrique, l'ordinateur exécute un autotest, ou **POST** ( Power On Self Test ). Vous le voyez au fait que la diode des lecteurs de disquettes s'allume un court instant et que la quantité de mémoire est décomptée devant vos yeux à l'écran.

## I/ AUTOTEST

Pendant cet autotest, les différents composants du système ( mémoire, carte graphique, clavier, lecteurs, ect... ) sont rapidement passés en revue et vérifié. Une erreur éventuelle est ainsi annoncée à l'écran et signalée par des bips sonores émis par l'ordinateur.

Si la carte graphique n'est pas en place, c'est le seul moyen de vérifier s'il existe un problème.

## II/ LE BIOS ET SES TACHES

Après l'autotest, le contrôle de l'appareil est aux mains d'une composante invisible et pourtant essentielle : Le **BIOS** ( Basic Input Output system )

Ce programme intégré dans une puce inaccessible à l'utilisateur est une **ROM** ( Read Only Memory ou mémoire morte ) et contrôle les fonctions de bases du PC et est conçu sur mesure pour le PC sur lequel il est installé. ( Où sur la même gamme du PC ). Il est conçu de façon à ce que le PC puisse fonctionner avec le système d'exploitation MS DOS.

Le BIOS peut être représenté comme un élément qui posséderait une prise à chaque extrémité. La première prise n'est pas normalisée et s'adapte à tous les PC de tous les constructeurs. L'autre prise, au contraire, est normalisée pour la connexion avec MS DOS.

Le BIOS est aussi responsable du fait que votre ordinateur est compatible IBM, c'est à dire qu'il est identique au PC original d'IBM sur le plan fonctionnel.

Le BIOS original d'IBM est protégé comme n'importe quel logiciel et n'a donc pas le droit d'être dupliqué.

Cependant, certaines sociétés ont développé un BIOS spécial en essayant d'imiter les fonctions du BIOS original sans en reproduire le code.

Les tout premiers ordinateurs annoncés comme étant compatible étaient tout sauf compatibles.

En 1985, la société PHOENIX réussit à développer légalement un BIOS fonctionnellement identique à celui d'IBM.

C'est ainsi que votre PC peut être considéré aujourd'hui réellement compatible.

Il existe aujourd'hui plusieurs sociétés constructrices de BIOS ( PHOENIX, AMERICAN MEGATRENDS... ) et de cela dépend vraiment votre compatibilité avec les autres micro-ordinateurs de type PC.  
Ces BIOS sont maintenant transformables avec des nouvelles versions.

Le BIOS est chargé des tâches de contrôle sur les fonctions élémentaires d'entrée et de sortie de l'ordinateur, mais aussi d'une tâche essentielle :

Il contient en effet un sous programme dont le rôle est de charger le système d'exploitation à chaque fois qu'on allume l'ordinateur.

Il existe deux façons de REBOOTER le PC :

- Par l'intermédiaire de l'interrupteur ( Démarrage à froid ou COLD BOOT )
- Par la combinaison des touches <Ctrl>+<Alt>+<Del> ( Démarrage à chaud ou Warm Boot )  
*<Suppr>*

Le BIOS est donc le premier élément à se rendre maître de l'ordinateur. Vous pouvez le constater au fait qu'un affichage du BIOS indique au départ le Nom du constructeur, avec le plus souvent accompagné d'un numéro de version avec l'année de production.

Sur certains ordinateurs, un autre message apparaît avant celui-ci, pour indiquer le constructeur de la carte vidéo et souvent la taille de la mémoire graphique ( VRAM ).

### III/ CHARGEMENT DU SYSTEME D'EXPLOITATION

Le système d'exploitation doit être chargé en mémoire principale à chaque fois qu'on allume l'appareil, car cette mémoire ( rapide mais très petite ) à un défaut lié à sa nature et à sa conception : ses données se volatilisent quand on éteint l'appareil. La mémoire du PC revient à son état initial : Elle se vide entièrement.

Le BIOS charge donc le système d'exploitation dans la mémoire principale. Cette opération obligatoire pour le fonctionnement du PC se fait pendant que le message :

*Lancement de MS-DOS...*

apparaît à l'écran.

La suite dépend ensuite de la configuration de votre SETUP. Soit un démarrage directement à partir de votre Disque dur soit une vérification de la présence d'une disquette système dans votre lecteur.

### IV/ QU'EST CE QU'UN SYSTEME D'EXPLOITATION

Le système d'exploitation MS DOS est fourni sur plusieurs disquettes contenant plusieurs Mo de données ( 1 octet équivaut à 1 caractère ). L'ensemble ne constitue pas réellement le système d'exploitation MS DOS. Il y a en plus des programmes auxiliaires et de compléments.

Le système d'exploitation proprement dit comprend trois éléments ou trois fichiers.

Les deux premiers sont des fichiers cachés : **IO.SYS** et **MSDOS.SYS** visible avec la **DIR /a** ou **DIR**,  
Le troisième est le fichier **COMMAND.COM** visible à l'écran par la commande **DIR**.

Un autre élément peut apparaître, le fichier **DBLSPACE.BIN** si vous avez comprimé votre disque dur avec le programme DBLSPACE.

Les deux fichiers cachés constituent le noyau du système d'exploitation. Ils doivent se trouver sur une partie spécial du disque dur ou de la disquette. Cette partie s'appelle le bloc de démarrage ( ou BOOT BLOCK ) du support de données. C'est pourquoi ces fichiers cachés ne peuvent pas être copiés sur le support de données avec les commandes de copies habituelles ( COPY ).

Il faut utiliser la commande **FORMAT /S** pour formater la disquette et installer les fichiers systèmes.

En revanche, le programme **COMMAND.COM** est la partie visible du système d'exploitation. Il représente ce qu'on appelle le processeur de commandes. Sa tâche est de recevoir vos commandes, de les interpréter et de les transmettre pour exécution ou même de les activer lui-même car il possède en lui tous les commandes essentielles du systèmes d'exploitation ( Les commandes internes ).

Tous les autres éléments du système d'exploitation sont les commandes externes et se trouvent sur le support de données ( Disque dur ) sous forme de fichiers indépendant et sont chargé au besoin en mémoire. Ces programmes externes sont importants pour travailler confortablement mais ne sont pas indispensable pour le fonctionnement du système d'exploitation.

S'il devait arriver que l'appareil se refuse à démarrer en affichant le message du type

### *Processeur de commandes défectueux ou inexistant*

Il suffit de copier à nouveau le fichier COMMAND.COM sur le support de données en question.

## **V/ LES FICHIERS CONFIG.SYS ET AUTOEXEC.BAT**

L'ordinateur n'est pas un appareil unique mais un ensemble de composants qui fonctionnent en collaboration avec l'ordinateur proprement dit soit l'ensemble Processeur / Mémoire.

Il n'y a pas d'équipement universellement valable pour le PC. Il faut spécifier l'ensemble de l'environnement et cette tâche est largement dévolue à l'utilisateur.

Le fait que vous soyez obligé de définir vous-même autant d'éléments pourrait apparaître comme un inconvénient de taille. Pourtant, cette ouverture du système a été l'une des raisons du succès du PC.

Le PC doit donc être réglé en fonction des divers composants intervenant dans le système. C'est ce réglage que l'on appelle la **CONFIGURATION**.

### **Il faut distinguer deux façons essentiellement différentes de configurer l'ordinateur.**

- Pour annoncer la présence des unités ( lecteur de disquette, disque dur ... ) dans le système, pour annoncer la quantité de mémoire principale disponible ainsi que d'autres paramètres du système, vous devez passer par le **SETUP** de l'ordinateur dont l'appel se réalise sur la plupart des ordinateurs en conservant la touche <Del> appuyée pendant le démarrage. Cette touche est indiquée lorsque le POST affiche son message :

*hit <Del> if you want to run SETUP*

Les réglages déposés dans le SETUP sont effectués en général une fois pour toutes et ne sont pas modifiés, sauf en cas de modifications des composants matériels ( nouveau disque dur , ajout de mémoire... ).

Ces valeurs sont gardées dans une mémoire spéciale, appelée CMOS-RAM, qui conserve les données même quand l'appareil est éteint, grâce à la présence d'une pile.

- L'autre mode de configuration de l'ordinateur est celui qui se renouvelle au contraire à chaque démarrage, car il faut charger des programmes de contrôle, qui s'effacent de la mémoire dès que l'on éteint l'appareil.

Cette configuration est réalisée grâce à deux fichiers spéciaux appelés  
CONFIG.SYS et AUTOEXEC.BAT

Ces deux fichiers ne sont pas indispensables mais très importants.

Les fichiers CONFIG.SYS et AUTOEXEC.BAT sont tous deux de purs fichiers texte, ne contenant que des chiffres et des lettres.

Ils peuvent être créés et modifiés avec l'éditeur **EDIT** de DOS.

Exemple : EDIT CONFIG.SYS

Ils sont aussi visualisés avec la commande **TYPE** de DOS.

Exemple : TYPE CONFIG.SYS

**Ces deux fichiers doivent se trouver dans le répertoire principal ( ou RACINE ) de l'unité de lancement.**

## CONFIGURATION MEMOIRE

La mémoire principale est aussi appelée RAM ( Random acces Memory ), qui pourrait se traduire par Mémoire à accès aléatoire ( Aléatoire dans le sens où il est possible d'accéder à n'importe quelle partie ).

Le processeur ou les programmes déposent leurs données dans cette mémoire. Ces données peuvent être à tout moment supprimées ou remplacées par d'autres éléments.

Avec le processeur, la mémoire principale est la composante essentielle du PC, puisque toutes les données nécessaires se trouvent en attente dans cette mémoire très rapide ( Temps d'accès : 80, 70, 60 Nanoseconde ) pendant le fonctionnement d'un programme.

Le système d'exploitation est chargé dans cette RAM au démarrage.

La même chose est vraie pour les gestionnaires de périphérique, qui contrôle les appareils connectés en collaboration avec le système d'exploitation lui-même.

Le processeur dépose donc ses données dans un emplacement de la mémoire principale.

Le dépôt des données en mémoire s'effectue de manière aléatoire. Il n'y a pas d'emplacement déterminé à l'avance. Le dépôt se fait simplement là où il y a de la place tout en conservant une trace de l'emplacement en mémoire de son contenu.

Il faut remarquer que comme dans tout système moderne de magasinage, les données sont retrouvées ensuite sans aucune hésitation, grâce au fait que chaque emplacement en mémoire à son adresse propre. ( En hexadécimal ).

### I/ BASE DE GESTION MEMOIRE

La taille de la mémoire principale est certes illimitée en théorie, elle est limitée par votre carte mère, mais vous ne pouvez utiliser en fait sous DOS que 640 Ko, même si votre mémoire fait au total 32 Mo. Tout le reste n'existe pas pour le DOS.

Cet état, étonnant au premier abord, s'explique aisément :

Le système d'exploitation DOS a été écrit pour le PC apparu sur le marché en 1981, avec le processeur 8088 d'Intel. Ces processeurs avaient alors des capacités gigantesques, par rapport aux processeurs de l'époque. Il disposait en particulier de 20 conduits d'adressage pour les données en mémoire. Ces 20 conduits constituaient ce qu'on appelle le bus d'adresse.

Avec 20 conduits d'adresse, sur un système qui utilise le système binaire, on ne pouvait cependant adresser que 2 puissance 20 octets, soit 1.048.579 octets ( 1Mo ), car comme dans tout ordinateur digital, chaque conduit ne peut transporter qu'un 0 ou un 1.

En 1981, on ne savait pas qu'il y aurait un jour des processeurs avec 32 voir 64 et 128 conduits d'adresse ( 32 Bits, 64 Bits, 128 Bits ). On a donc conçu tout le système d'exploitation en fonction du processeur existant.

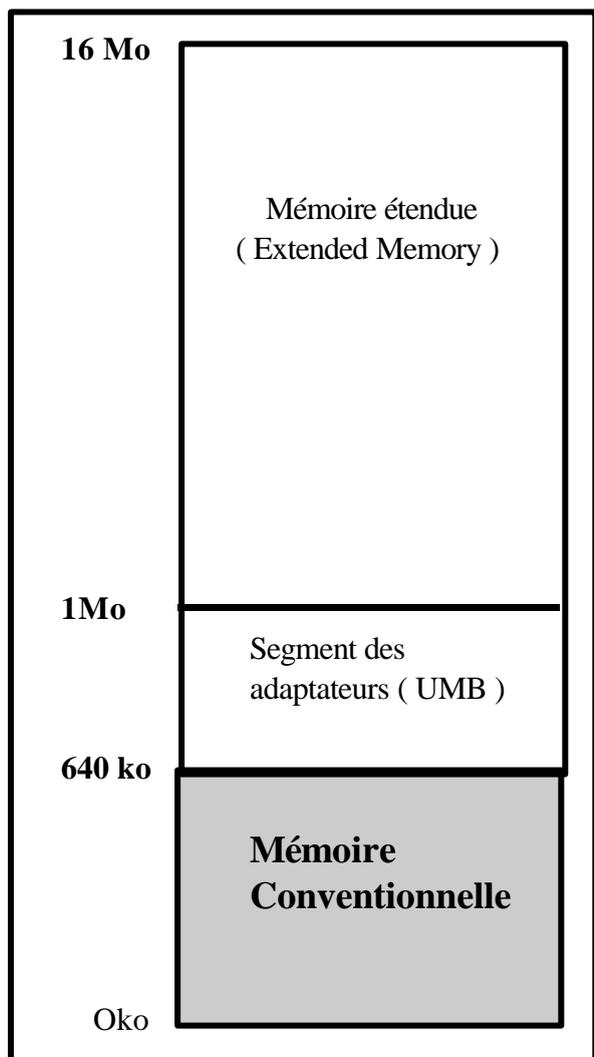
Si l'on sait de plus qu'IBM avait réservé pour ses besoins propres tout le secteur s'étendant de 640 Ko à 1Mo ( 1024 Ko ), on voit pourquoi une telle situation est devenue ensuite inconfortable sur des ordinateurs qui disposent, eux, de 8, 16, 32... Mo, alors que le système d'exploitation ne peut pas utiliser plus que 640 Ko.

Ce secteur utilisable sans restriction par tous les programmes est appelé **MEMOIRE CONVENTIONNELLE**.

On peut se demander alors : Pourquoi construit-on donc des ordinateurs avec tant de mémoire, s'il est vrai que le système d'exploitation ne peut pas adresser une grande partie de cette mémoire ? A quoi sert-elle ?

La réponse est simple : elle est quand même utilisée, même sous DOS mais d'une manière différente.

## DISTRIBUTION DE LA MEMOIRE SOUS MS DOS :



## II/ LES DIFFERENTS TYPES DE MEMOIRE

### 1°/ La mémoire paginée ( Expanded Memory ) : Mémoire EMS

Pour utiliser la mémoire au delà de cette frontière magique de 1Mo, on a d'abord eu l'idée d'offrir à l'ordinateur de la mémoire sur une carte d'extension ( la célèbre **Above Board** d'Intel ), sous la forme d'une mémoire paginée. On la nomma alors **Expanded Memory Specification** ou **EMS**.

### La mémoire paginée se nomme aussi mémoire EMS.

Sur les ordinateurs du type i386 et au-dessus, on n'a plus besoin de ces cartes spéciales d'extension mémoire. On peut installer directement la mémoire d'extension existante au-dessus de 1Mo sous forme de mémoire paginée. Le gestionnaire fourni sous DOS pour gérer la mémoire paginée s'appelle **EMM386.EXE**.

Cette mémoire ne peut pas être utilisée par tous les logiciels, aussi, on utilise pas cette mémoire pour toutes les applications WINDOWS.

Tous les logiciels utilisant cette mémoire vous le signalent dans leur documentation.

### 2°/ Mémoire étendue ( Extended Memory ) : Mémoire XMS

En 1988, Microsoft présente, en accord avec d'autres sociétés productrices de logiciels, le standard XMS ( Extended Memory Specification ), destiné à régler et l'utilisation de cette mémoire. A partir de là, un logiciel conforme à ce standard pouvait utiliser l'extension comme si c'était une mémoire conventionnelle.

On l'appelle alors la mémoire étendue. La condition pour cela était tout de même la présence d'un gestionnaire chargé de veiller à ce que les règles soient strictement observées.

Le gestionnaire DOS chargé de la gestion de ce secteur en mémoire s'appelle **HIMEM.SYS**.

### 3°/ Secteur UMB ( Upper Memory blocks )

A l'époque où chaque zone de 10 Ko récupérée en mémoire était fêtée comme une conquête, les tentatives n'ont bien sûr pas manqué pour fournir autant d'espace que possible aux applications si gourmandes en mémoire.

On s'est donc penché avec attention sur la zone située entre 64 Ko et 1Mo ( 1024 Ko ), réservé par IBM pour ses propres usages et en principe inconnue au système d'exploitation MS DOS.

Les logiciels de contrôle sont chargés dans cette zone. Ces logiciels sont nécessaires aux composantes du Hardware. C'est pourquoi la zone en question est appelée segment d'adaptateur.

Il existe un segment dans lequel a été créée la mémoire de la carte graphique utilisée. Cette zone comprend une partie prévue pour une carte graphique couleur et une autre pour une monochrome.

Puisque l'on a rarement besoin de plus d'une carte graphique, l'une de ces deux secteurs de 32 Ko reste toujours libre. D'autres zones restées de même en jachère, de sorte que l'on parvient à récupérer de la sorte environ 120 Ko de mémoire, en détournant de leur finalité initiale. Ces blocs de mémoire sont appelés

**Upper Memory Block : UMB**

Depuis la version 5.0 de DOS, sur un ordinateur muni au moins d'un processeur i386, il est possible de déposer dans cette zone de mémoire des programmes résident et des gestionnaires de périphériques, ou même des parties du système d'exploitation, ce qui permet d'alléger d'autant la mémoire conventionnelle.

Le gestionnaire EMM386.EXE, qui installe et gère normalement la mémoire paginée a aussi la tâche de vous ouvrir l'accès aux UMB et de mettre ainsi plus de mémoire à la disposition de vos applications.

Le dépôt du système d'exploitation dans les UMB est réalisé avec la commande DOS, placée dans le CONFIG.SYS :

***DOS=UMB***

### 4°/ Zone de mémoire haute ( HMA )

La maladie héréditaire introduite par MS DOS dans le système PC a donc pour origine le fait que le système d'exploitation a été écrit pour le processeur 8088, qui ne pouvait adresser que 1Mo de mémoire principale, à cause de ses 20 conduits d'adresse.

A cause d'une particularité de l'adressage de la mémoire, le dernier emplacement adressable ne se situe pas exactement à 1Mo ( 1024 Ko ), mais à 1072 Ko. On dispose donc de 48 Ko supplémentaire.

Depuis la version 5.0 de DOS, avec un ordinateur équipé au moins d'un processeur i286, vous pouvez déposer une partie du système d'exploitation dans cette zone appelée **Zone de mémoire haute**, ou **High Memory Area ( HMA )**, et gagner ainsi 48 Ko de mémoire conventionnelle.

Pour déposer le système d'exploitation dans la HMA au démarrage, utilisez dans CONFIG.SYS la commande déjà rencontré :

***DOS=HIGH***

Remarque : Si on veut également utiliser les UMB, on utilise la commande suivante :

***DOS=HIGH,UMB***

En résumé, lorsqu'on parlera de la RAM utilisée sous MS DOS, on parlera :

- **Mémoire Conventionnelle**
- **Mémoire XMS**
- **Mémoire EMS**
- **Mémoire HMA**
- **Mémoire UMB**

Pour vérifier où sont placés vos gestionnaires dans cette Ram, utiliser la commande **MEM**.

## I/ TACHES DES FICHIERS DE CONFIGURATION

C'est un fait remarquable que l'automatisation du démarrage exige deux fichiers, ayant apparemment des tâches différentes.

A savoir : **AUTOEXEC.BAT** et **CONFIG.SYS**

La différence entre ces deux fichiers et les tâches respectives qui leur sont dévolues est effectivement considérable. La différence entre les deux fichiers tient tout d'abord dans le fait que CONFIG.SYS est toujours recherché en premier pour être traité ligne par ligne.

Dans **CONFIG.SYS**, on place des commandes bien déterminées destinées à charger des composantes logicielles. Les programmes chargés par l'intermédiaire de CONFIG.SYS vont en général très loin dans le contrôle du Hardware. On nomme ces programmes des **Gestionnaires de Périphériques**, car ils rendent possible le fonctionnement de certains périphériques dans une position privilégiée à l'intérieur de la mémoire principale ( d'où l'importance de l'ordre dans lequel ils sont placés ).

Les programmes chargés grâce au CONFIG.SYS doivent être d'un type bien particulier, ils ne sont pas lancés comme pourrait être un programme quelconque. Certains gestionnaires de périphériques peuvent dépendre l'un de l'autre ou s'exclure mutuellement. L'ordre des gestionnaires et des commandes dans CONFIG.SYS ne peut donc pas être arbitraire.

Dans **AUTOEXEC.BAT**, en revanche, vous pouvez lancer automatiquement n'importe quel programme se trouvant sur le disque dur. Ce fichier permet donc d'automatiser des processus répétitifs.

Vous pouvez par exemple taper vous-même après le démarrage le programme KEYB.COM correspondant à la gestion du clavier, mais vous pouvez aussi demander à AUTOEXEC.BAT de s'en occuper de façon automatique. Puisque les programmes qui interviennent dans AUTOEXEC.BAT sont des programmes habituels, leur position à l'intérieur du fichier importe peu à part quelques-uns.

En revanche, les gestionnaires de périphériques qui interviennent dans CONFIG.SYS ne peuvent pas être lancés autrement qu'à partir de ce fichier.

Les spécifications effectuées par l'intermédiaire de CONFIG.SYS restent valide jusqu'à l'extinction de l'ordinateur et ne peuvent être modifiées entre temps que de manière exceptionnelle. En revanche, les réglages dus à AUTOEXEC.BAT ne sont pas nécessairement prévus pour demeurer en mémoire jusqu'à l'extinction de l'appareil.

La dernière différence : Le fichier CONFIG.SYS ne peut être lancé que par le PC.

Le fichier AUTOEXEC.BAT, lui, est certes lancé par l'ordinateur au moment du démarrage, ( AUTOEXEC = exécuté automatiquement ), mais il pourrait théoriquement être lancé par vous-même n'importe quand, puisqu'il s'agit d'un fichier de type BATCH normal.

## II/ GESTIONNAIRE DE PHERIPHERIQUE

La raison qui explique l'extraordinaire succès du PC est que le contrôle des appareils externes soit exécutée par des gestionnaires déposés à l'intérieur du système.

Sur les gros ordinateurs, par exemple, le contrôle des appareils connectés au processeur de manière interne ou externe est dans ce cas intégré au système d'exploitation, les modifications sont très difficiles à réaliser.

Au contraire, la conception de MS DOS était différente dès le début : le Système d'exploitation ne contrôle pas lui-même les appareils périphériques. Il possède une sorte d'entrée normalisée, dans lequel des programmes - précisément ces gestionnaires de périphériques ou **DRIVERS** - viennent se greffer, ce qui permet de connecter le nouveau périphérique au système d'exploitation.

Même les composants qui font partie intégrante du PC ne sont pas contrôlés directement par le système d'exploitation, mais possèdent au contraire chacun leur gestionnaire, chargé à lui seul du contrôle.

L'avantage de ce système est évident : on peut connecter n'importe quel appareil, à une seule condition : Il doit apporter avec lui son gestionnaire, conforme aux conventions de DOS et donc adapté au système d'exploitation.

Tant que les développeurs respectent les règles qui régissent la connexion d'un périphérique, ils peuvent toujours créer de nouveaux appareils sous MS DOS pour le PC.

Nous devons distinguer en gros deux types de gestionnaires de périphériques :

- Les gestionnaires de périphérique par caractères
- Les gestionnaires par blocs

### Les gestionnaires de périphérique par caractère

Les gestionnaires par caractère établissent la communication caractère par caractère ( octet par octet ) avec le Hardware connecté. Il faut compter parmi ceux-ci le gestionnaire CON, qui contrôle le clavier et l'écran octet par octet. Chaque pression sur une touche du clavier génère le caractère voulu sur l'écran. Le gestionnaire PRN, prévu pour l'imprimante connectée à la première interface parallèle contrôle lui aussi l'imprimante caractère par caractère, c'est à dire octet par octet.

Un gestionnaire par caractère est toujours responsable par sa conception même, d'un appareil déterminé. Il faut donc dans ce cas un gestionnaire de périphérique par appareil.

### Les gestionnaires de périphériques par bloc

Les gestionnaires par bloc se comportent différemment. Puisqu'ils servent à la communication du système d'exploitation avec les mémoires de masse comme le disque dur ou le lecteur de disquette, le transfert ne se fait pas par caractère mais par blocs entiers, dont la taille peut varier en fonction de l'appareil ou sur le même appareil en fonction des circonstances.

Les noms des gestionnaires par blocs correspondent en général aux noms des lecteurs, soit A : pour le premier lecteur de disquette ou C : pour le premier disque dur.

Puisque la structure de la communication pour un disque dur C : peut être analogue ou identique à celle d'une unité D : ; il n'est pas nécessaire d'avoir un gestionnaire par blocs différent pour chaque périphérique.

A coté de la distinction entre les gestionnaires de périphérique par caractères et par blocs, on distingue également des **GESTIONNAIRES INTERNES** ( par exemple CON pour le clavier et l'écran ) intégré à demeure dans le système d'exploitation, que l'on ne peut ni charger ni supprimer, et des **GESTIONNAIRES EXTERNES** ( par exemple KEYBOARD.SYS pour la spécification du clavier ou COUNTRY.SYS pour la définition des données propres à un pays, comme le format de date ).

Ces derniers ne sont pas chargés automatiquement par MS DOS au démarrage du PC, mais il faut les placer dans le CONFIG.SYS.

### GESTIONNAIRES INTERNES

Les gestionnaires de périphériques interne du système d'exploitation sont chargé automatiquement au démarrage par l'intervention du BIOS. Vous n'avez donc pas à vous en occuper vous-même.

L'ordre est ici le suivant :

PERIPHERIQUE	UTILISATION
NUL	Périphérique <i>dummy</i> équivalent à un trou noir
CON	Gestionnaire pour le clavier et l'écran
AUX	Première interface série
PRN	Première interface parallèle
CLOCK\$	Gestionnaire de l'heure système
A :	Gestionnaire de premier lecteur de disquettes
C :	gestionnaire de premier disque dur
LPT1 à 3	Première à troisième interface parallèle
COM1 à 4	Première à quatrième interface série

Les noms de ces gestionnaires de périphérique sont réservés et ne doivent donc pas être utilisés à d'autres fins. En particulier, ils ne doivent pas intervenir dans un nom de fichier ou de répertoire.

Au contraire, les gestionnaires de périphérique externes ne sont pas chargés automatiquement. Il faut faire intervenir la commande `DEVICE` dans `CONFIG.SYS` pour les charger en mémoires.

Si vous chargez un gestionnaire externes par l'intermédiaire de `CONFIG.SYS`, par exemple le gestionnaire de clavier `ANSI.SYS` avec la ligne :

`DEVICE=ANSI.SYS`

Le chargement se fait après le périphérique interne `NUL`, mais avant tout les autres, l'ordre étant le suivant :

NUL  
ANSI  
CON  
AUX  
PRN  
CLOCK\$

...

La raison en est claire. Chaque gestionnaire de périphérique doit pouvoir remplacer un autre gestionnaire dans toutes ses fonctions, en se chargeant par exemple de ses tâches de manière plus complète et plus adaptée.

Dans notre cas, `ANSI.SYS` ( un gestionnaire de périphérique ajouté ultérieurement, mais plus adapté au contrôle de l'écran et du clavier ) remplace le gestionnaire interne `CON`, un peu restreint.



## ■ **COUNTRY.SYS** \*

### Spécification de pays

Ce gestionnaire de périphérique permet de spécifier les données propres au pays dans lequel on se trouve comme le format de date, le séparateur décimal ou le format d'heure. Sans ce gestionnaire, c'est le format américain qui est utilisé.

Il faut toutefois intégrer ce gestionnaire à l'aide d'une commande qui lui est propre :

```
COUNTRY=033,,C : \DOS\COUNTRY.SYS
```

033 pour la France

Le gestionnaire de périphérique COUNTRY.SYS n'est donc pas intégré directement avec la commande DEVICE.

## ■ **DBLSPACE.SYS** \*

### Lecteur comprimé

Si vous avez comprimé votre disque dur avec le programme DBLSPACE, vous pouvez utiliser ce gestionnaire afin de définir la position en mémoire principale du gestionnaire DBLSPACE.BIN.

Voir le programme DBLSPACE.EXE

## ■ **DISPLAY.SYS** \*

### Tableau des polices de caractères

Ce gestionnaire d'écran génère des tableaux de polices de caractères, qui n'ont cependant pas besoin en temps normal d'être définis explicitement, puisque la police standard suffit amplement pour notre alphabet.

```
DEVICE=C : \DOS\DISPLAY.SYS
```

## ■ **DRIVERS.SYS** \*

### Lecteur de disquette

Il s'agit d'un gestionnaire supplémentaire de lecteur de disquette, annonçant et contrôlant des appareils externes ( par exemple, un troisième lecteur de disquette ), pour faire fonctionner de nouvelles unités sous forme de lecteurs logiques, quand leur nombre dépasse le nombre d'unité standard. ( 2 )

```
DEVICE=C : \DOS\DRIVER.SYS {option}
```

Les options sont à rajouter permettant de définir de plus près le lecteur à annoncer.

## ■ **EGA.SYS**

### Dos-shell

Gestionnaire en relation avec le programme DOS SHELL sous DOS. Intégrer ce gestionnaire dans le CONFIG.SYS si votre système fonctionne avec une carte graphique de type EGA. Sinon ce gestionnaire est inutile.

## ■ **EMM386.EXE**

### Mémoire paginée

Voici un gestionnaire de périphérique important. LA mémoire principale au-dessus de 1Mo peut être rendue disponible grâce à lui sous forme de mémoire paginée pour les applications. Afin d'installer et de gérer cette mémoire d'extension comme mémoire paginé, vous avez besoin de EMM386.EXE.

Les blocs de mémoire normalement inaccessibles même aux programmes d'application dans le secteur qui s'étend entre 640 Ko et 1 Mo peuvent être utilisés à certaines fins grâce à ce gestionnaire, ce qui permet de décharger un peu la mémoire principale ( conventionnelle ), où la place est étroite.

La syntaxe pour intégrer EMM386.EXE dans le CONFIG.SYS est la suivante :

*DEVICE=C :\DOS\EMM386.EXE {options}*

Les options sont très variées et le danger d'une configuration erronée est d'autant plus grand. Voici les options essentielles :

PARAMETRES	FONCTION
Taille	Indique la taille de la mémoire à utiliser en tant que mémoire paginée. La valeur standard est de 256 Ko. Les valeurs possibles sont entre 16 et 32.768 Ko.
RAM=secteur	Permet de rendre disponible non seulement la mémoire paginée mais aussi les blocs de mémoire supérieure ( UMB ). Si vous n'indiquez rien en ce qui concerne le secteur que le gestionnaire doit utiliser pour les UMB ; l'ensemble de la mémoire disponible est mis à contributions.
NOEMS	Cette option demande à EMM386 de mettre à votre disposition uniquement les blocs de mémoire supérieure. L'accès à la mémoire paginée n'est pas possible. Conseillé si vous travaillez essentiellement sous Windows.

Notez bien que pour utiliser ce gestionnaire dans CONFIG.SYS, vous devez libérer auparavant le secteur de la mémoire étendue, donc la mémoire au-dessus de 1Mo, en vous servant du gestionnaire HIMEM.SYS ( voir plus loin ). Il faut donc que la ligne d'intégration du EMM386.SYS se trouve au delà

La ligne

*DEVICE=C :\DOS\EMM386.SYS 1024 RAM*

installe 1 Mo ( 1024 Ko ) de mémoire paginée et met en place les blocs de mémoire supérieurs ( UMB ).

Si vous voulez apporter ultérieurement des modifications dans l'exécution, vous devez écrire la commande suivante sous DOS :

*EMM386 {Paramètre}*

Les paramètres sont les suivants :

- **ON**                    Active le gestionnaire déjà chargé par CONFIG.SYS
- **OFF**                Désactive le gestionnaire chargé. Ce n'est toutefois possible que s'il n'y a pas de mémoire paginée en exercice et si les UMB ne sont pas installés.
- **AUTO**            la mémoire paginée n'intervient que si le programme l'exige.

■ **HIMEM.SYS**

Mémoire étendue

La mémoire principale au-dessus de 1Mo n'est pas reconnue par le système d'exploitation MS DOS et ne peut donc pas être utilisée en temps normal. Le gestionnaire HIMEM.SYS permet d'installer et de gérer cette mémoire sous forme de mémoire étendue ( EXTENDED MEMORY ).

A cette condition seulement, les programmes ( par exemple Windows ) peuvent accéder à cette mémoire et la mettre à contribution. Si le gestionnaire EMM386.EXE veut ensuite revendiquer une partie de celle-ci sous forme de mémoire paginée ( EXPANDED MEMORY ), il faut d'abord que HIMEM.SYS soit intervenu pour la rendre accessible.

*DEVICE=C:\DOS\HIMEM.SYS*

■ **INTERLNK.EXE \***

Communication par interface série

Ce gestionnaire de périphérique contrôle les allées et venues entre ordinateurs reliés par l'intermédiaire de l'interface série, pour transporter des données d'un appareil à l'autre. Un câble spécial est nécessaire pour le transfert. Il s'agit du câble de MODEM NUL.

*DEVICE=C:\DOS\INTERLNK.EXE {option}*

INTERLNK est utilisé d'une part comme gestionnaire de périphérique avec la commande DEVICE dans CONFIG.SYS et d'autre part sous forme de commande normale pour relier entre eux les lecteurs de plusieurs ordinateurs grâce à la commande :

*INTERLNK [client=serveur]*

Les options les plus importantes sont les suivantes :

PARAMETRES	FONCTION
/lect :n	Nombre de lecteurs sur le serveur devant être reliés au client qui le demande. Le nombre par défaut est 3.
/COM :n	Numéro de l'interface série par l'intermédiaire de laquelle on veut créer la liaison. Sans indication de numéro, c'est la première interface qui est utilisée.
/LPT :n	Comme /COMM :n mais pour l'interface parallèle.
/NOPRINTER	Aucun connecteur d'imprimante n'est relié au client.
/AUTO	INTERLNK ne sera installé que si une liaison avec le serveur est trouvée. Est normalement toujours installé.
/NOSCAN	Le gestionnaire est installé, mais aucune liaison n'est établie pour le moment avec le serveur.
/BAUD :vitesse	La vitesse en bauds pour la liaison série est spécifiée par cette option. La spécification standard est de 115200 bauds. Si l'on veut utiliser une autre valeur, il faut veiller à ce que la même valeur soit spécifiée sur le serveur avec INTERSRV.

Pour relier un lecteur et utiliser pour le reste les spécifications par défaut, vous, utiliserez la ligne suivante dans le CONFIG.SYS

*DEVICE=C :\DOS\INTERLNK.EXE /Lect :I*

Les paramètres pour la commande sous DOS :

- **Client :** Le lecteur de l'ordinateur qui doit être dérivé vers un lecteur du serveur Interlnk.
- **Serveur :** Le lecteur sur le serveur, vers lequel le client doit être dérivé. Ce lecteur doit être l'un des lecteurs indiqués comme dérivés sur l'écran du serveur. Sans indication de serveur, une liaison existante est dissoute.

■ **POWER.EXE \*** Surveillance du courant

Ce gestionnaire surveille l'alimentation en courant sur les portables, où le système est gelé lorsqu'aucune activité n'est constatée de la part de l'utilisateur. Cela permet d'économiser les ressources de la batterie.

*DEVICE=C :\DOS\POWER.EXE ADV :{option}*

Le gestionnaire est chargé en mémoire par l'intermédiaire de la commande DEVICE dans CONFIG.SYS. Vous avez ensuite la possibilité de faire passer la surveillance de l'alimentation en d'autres modes à l'aide de la commande POWER sous DOS.

La commande est la suivante :

*POWER {option }*

**Les options sont les suivantes :**

- **MAX :** Economie maximale de courant, influant sur la vitesse de fonctionnement des applications.
- **REG :** Spécification par défaut, qui ne concerne pas en général les applications.
- **MIN :** Vous devrez utiliser cette option quand la durée d'utilisation d'une application n'est pas satisfaisante. Cela réduit toutefois l'économie de courant.
- **OFF :** Désactive le contrôle du courant mais laisse le gestionnaire en mémoire. Par l'appel renouvelé de la commande, on peut activer à nouveau la commande.

Pour demander la plus grande économie de courant dès le démarrage, vous placerez dans CONFIG.SYS la ligne suivante :

*DEVICE=C :\DOS\POWER.EXE ADV :MAX*

Si vous voulez revenir ultérieurement à une position moyenne, vous ferez intervenir la commande :

*POWER ADV :REG*

■ **RAMDRIVE.SYS \*** Lecteur en RAM

Un disque RAM est un lecteur simulé en mémoire principale, dont l'accès se fait pour cette raison à très grande vitesse. En contrepartie, ce lecteur présente deux inconvénients de taille : Il encombre la mémoire et son contenu n'est pas inscrit à demeure dans celle-ci, puisqu'il se volatilise quand on éteint l'appareil. On crée un disque RAM avec RAMDRIVE.SYS.

*DEVICE=C :\DOS\RAMDRIVE.SYS {option}*

Les options sont les suivantes :

PARAMETRES	FONCTION
Taille	Capacité du disque RAM en Ko. Sans aucune indication, la taille est de 64 Ko. La valeur min est 4 Ko, sa valeur Max est de 31744 Ko.
Secteur	Définit la taille des secteurs simulés. Valeur Standard : 128. Si vous modifiez cette valeur, vous devez modifier aussi la taille du lecteur.
Nombre	Définit le nombre de fichiers pouvant être créé sur le disque RAM. Valeurs autorisées : 2 à 512. Valeur standard : 64
/E	Si vous disposez d'une mémoire étendue, vous pouvez créer un disque RAM au delà de la limite de 1 Mo en vous servant de cette Option.
/A	Création du disque RAM dans la mémoire paginée.

■ **SETVER.EXE** \* Emulation de version MS DOS

Il arrive toujours que certaines applications demandent au système d'exploitation son numéro de version, pour s'assurer que la version est suffisamment récente. SETVER.EXE répond à l'interrogation de façon à faire croire au programme que le numéro de version est inférieur au numéro réel.

*DEVICE=C:\DOS\SETVER.EXE*

### 3°/ GESTIONNAIRES STATIQUES ET DYNAMIQUES

Vous avez remarqué que certains gestionnaires portaient l'extension .SYS et d'autres l'extension .EXE, comme des programmes normaux.

La raison de cette différence tient au fait que certains gestionnaires sont chargés et restent en mémoire jusqu'à ce qu'on éteigne l'appareil. Ce sont ceux qui sont caractérisés par leur extension .SYS. Ces gestionnaires répondent à la conception ( un peu vieille ) de MSDOS, et ils n'ont pas été modifiés depuis le début.

Pour les gestionnaires qui ont pour extension .EXE, ils représentent des spécimens plus récents. Ils peuvent être modifiés dans le temps pendant un travail sous MSDOS.

Exemple, le gestionnaire POWER.EXE.

### 4°/ COMMANDES DE CONFIG.SYS

Indépendamment de l'intégration des gestionnaires de périphériques de la manière décrite, il existe toute une série de commande permettant de définir certains paramètres systèmes ou de modifier des paramètres en vigueur dans le PC.

Pour ces commandes, vous n'avez pas besoin de DEVICE. Elles interviennent suivant la syntaxe suivante :

## COMMANDE=Paramètres

le nom de la commande est suivi d'un signe d'égalité puis de l'option souhaitée.

### ■ **BREAK** Autoriser l'interruption

Sur un PC, vous disposez d'une combinaison de touches immuable, à l'aide de laquelle vous pouvez interrompre une commande DOS et certains programmes :

<Ctrl>+<C>

Exemple : Lorsque vous avez par exemple appelé la commande DISCOPY et que l'ordinateur réclame des disquettes vides alors que vous n'en avez pas sous la main, vous pouvez annuler l'opération avec ses combinaisons de touches.

Toutefois, cette annulation ne fonctionne que s'il n'y a pas d'accès en cours en lecteur. Si vous avez déjà introduit des disquettes et si le processus de copie a commencé, vous ne pouvez interrompre l'opération qu'au moment où le programme demande l'introduction d'une nouvelle disquette. En revanche, si vous avez placé dans CONFIG.SYS la ligne :

*BREAK=ON*

une vérification continue a lieu, même pendant l'accès à un lecteur, pour voir si vous n'avez pas actionné la combinaison de touches.

Vous pouvez ainsi interrompre la fonction à tout moment.

### ■ **BUFFERS** Tampons et vitesse d'accès

Quand vous ouvrez un fichier, par exemple un texte issu de votre traitement de texte, c'est les 512 premiers caractères qui sont lus d'un seul coup et placés dans un mémoire tampon. Si vous voulez modifier ce fichier, l'ordinateur examine d'abord si le caractère modifié ne se trouve pas dans l'un des tampons. En effet, l'accès à une mémoire tampon se fait beaucoup plus rapidement que l'accès au disque dur qui est en comparaison très lent. Si le caractère ne se trouve pas dans un tampon, mais dans ce cas seulement, le texte correspondant est chargé à partir de la mémoire de masse.

Le nombre des mémoires tampons aménagés accélèrent donc l'accès aux données. Cette commande a perdu aujourd'hui de son importance, car la mémoire cache fournit un moyen beaucoup plus efficace d'accélérer l'accès aux données. Pourtant, même sur un système qui dispose d'une mémoire cache, vous devez aménager quelques tampons, car beaucoup de programmes d'application en font encore usage.

Pour créer 20 Tampons, ce qui est une valeur raisonnable et courante dans la pratique, vous devez entrer dans le CONFIG.SYS la ligne suivante :

*BUFFERS=20*

Sur les systèmes disposant d'un cache, vous pouvez réduire cette valeur à 2.

## ■ **COUNTRY**

Données nationale : Pays

La commande COUNTRY permet de définir les données nationales spécifiques, comme le format de date et d'heure ou la nature du séparateur décimal. Elle se fait à l'aide du gestionnaire COUNTRY.SYS et des informations qui y sont déposées. Il suffit pour cela d'indiquer le pays voulu, par l'intermédiaire de son code téléphonique international, 033 pour la France, 001 pour les états unis.

La ligne du CONFIG.SYS sera :

*COUNTRY=033,,C : \DOS\COUNTRY.SYS*

## ■ **DOS**

Chargement dans les UMB, HIGH

Le système d'exploitation DOS est chargé de manière standard dans la mémoire principale conventionnelle. Si vous disposez d'un ordinateur ayant au moins un processeur i286, vous pouvez utiliser la mémoire située directement au-dessus de 1 Mo pour y déposer une partie du système d'exploitation et décharger de cette façon la mémoire conventionnelle. Pour faire sortir des parties du système d'exploitation de la mémoire conventionnelle, vous placerez dans CONFIG.SYS la ligne

*DOS=HIGH*

Le secteur de mémoire situé entre 640 Ko et 1 Mo, normalement inaccessible, peut être lui aussi utilisé pour le dépôt du système d'exploitation, ce qui permet là encore d'économiser de la place dans la mémoire conventionnelle.

Si vous avez libéré les blocs de mémoires supérieurs ( UMB ) avec le gestionnaire EMM386.EXE, vous pouvez utiliser ce secteur :

*DOS=HIGH,UMB*

Cette commande doit intervenir après libération de la mémoire étendue par HIMEM.SYS ou des blocs de mémoire supérieure par EMM386.EXE.

## ■ **DRIVPARM**

Modifications des lecteurs

Si vous désirez modifier les paramètres standard d'un lecteur, utilisez cette commande et indiquez les nouveaux paramètres avec lesquels vous voulez faire fonctionner le lecteur concerné. Cette commande sera en réalité que très rarement utilisée.

Vous disposez d'un lecteur 1,44Mo, mais vous utilisez uniquement des disquettes 720Ko, utilisez la commande suivante :

*DRIVPARM=/D :1 /F :2*

Au prochain démarrage, le lecteur de disquettes sera traité comme un lecteur 720 Ko.

## ■ **FCBS**

Blocs de contrôle des fichiers

Une méthode maintenant démodée pour contrôler les fichiers ouverts simultanément en mémoire consistait à utiliser des blocs de contrôle des fichiers ( File Control Blocks ). Le nombre maximal de blocs est défini au moyen de cette commande.

## ■ **FILES**

Fichiers ouverts

FILES permet de définir le nombre maximal de fichiers pouvant être ouvert simultanément en mémoire. Le nombre par défaut est de 8, ce qui est cependant bien trop faible. Une bonne valeur confirmée par la pratique est de 20. Avec les bases de données, il peut cependant s'avérer nécessaire de porter cette valeur à 50 ou plus.

*FILES=50*

## ■ **INSTALL**

Programmes résidents

Les programmes résidents en mémoire sont appelés normalement par AUTOEXEC.BAT. Si vous les chargez dans CONFIG.SYS, on y gagnera en mémoire mais ils ne pourront pas se placer hors de la mémoire conventionnelle, d'où son inutilité aujourd'hui.

## ■ **INSTALLHIGH**

Chargement en mémoire supérieur

Cette commande non mentionnée dans la documentation charge le programme résident comme INSTALL, mais dans la mémoire supérieure et non dans la mémoire principale conventionnelle. Il existe des avantages, mais on ne l'utilise que très rarement, car le programme MEMMAKER ne fonctionne pas avec ce procédé.

## ■ **LASTDRIVE**

Indication du dernier lecteur possible

Vous pouvez utiliser par défaut jusqu'à cinq lecteurs, jusqu'à la lettre E. Puisque c'est insuffisant quand vous partitionnez votre disque dur où lorsque vous travaillez dans un réseau, vous pouvez indiquer au système d'autoriser plus de lecteur que ceux prévus de manière standard. Il suffit pour cela d'indiquer le lecteur autorisé en dernier ressort :

*LASTDRIVE=Z*

Cette commande permet d'utiliser des lecteurs jusqu'à la lettre Z. Songez toutefois qu'un emplacement en mémoire est réservé pour chacun d'eux. Donc il vaut mieux indiquer les lecteurs dont vous avez réellement besoin.

## ■ **NUMLOCK**

Désactiver <NUM>

Avec les BIOS modernes, la touche <NUM> est activée après le démarrage. Le bloc numérique qui se trouve à droite du clavier est donc actif.

*NUMLOCK=ON*

Active au démarrage le bloc numérique

*NUMLOCK=OFF*

Désactive au démarrage le bloc numérique

## ■ **REM**

Commentaire

Il peut être très utile de placer de temps en temps une ligne de commentaire dans le fichier CONFIG.SYS pour se rappeler la signification de tel ou tel gestionnaire. Vous pouvez aussi désactiver une ligne du CONFIG.SYS ou de l'AUTOEXEC.BAT en la faisant précéder de REM

*REM DEVICE...*

Au prochain démarrage, ce gestionnaire ne sera pas chargé.

## ■ SHELL

### Processeurs de commandes

SHELL est conçu pour charger un processeur de commandes différents de COMMAND.COM ou pour préciser le chemin d'accès à celui-ci s'il ne se trouve pas dans la racine. Vous pouvez l'utiliser toutefois afin de réserver plus de mémoire pour le processeur de commandes et les tâches qu'il doit accomplir. Cette mémoire qui se situe dans l'environnement immédiat du processeur de commandes est aussi appelée **mémoire d'environnement**. ( Voir chapitre suivant ).

Pour intégrer COMMAND.COM dans le répertoire C :\DOS comme le processeur de commande valide pour lui affecter en même temps 1024 octets de mémoire d'environnement, introduisez la ligne suivante :

```
SHELL=C :\DOS\COMMAND.COM C :\DOS /P /E :1024
```

## ■ STACKS

### Piles systèmes

Les piles sont des secteurs en mémoire qui sont nécessaires au système pendant l'exécution d'un programme. Un accès aux spécifications par défaut ne s'impose pas en général. Lors de son installation, WINDOWS insère dans le fichier CONFIG.SYS de beaucoup de systèmes une ligne contenant cette commande :

```
STACKS=9,256
```

Si vous désactivez cette ligne ( REM ), vous économiserez un peu de mémoire. Par contre, si vous voyez apparaître un message indiquant un débordement de pile, vous devrez la réactiver au plus vite.

## ■ SWITCHES

### Paramètre particulier

A partir de DOS4.0, le système d'exploitation supporte le clavier étendu, donc aussi les touches de fonction <F11> et <F12>, ce qui conduit toutefois à des problèmes avec les anciennes applications. Puisque ces programmes ne savent pas ce qu'est un clavier moderne, on voit surgir des erreurs. En plaçant la commande

```
SWITCHES=/F
```

dans le CONFIG.SYS, le message

*démarrage de MS-DOS...*

n'apparaîtra pas. Il sert en fait de pause pour vous permettre de provoquer un démarrage sélectif du système avec les touches <F5> et <F8>. L'option /N interdit entièrement l'usage des touches F5 et F8.

## FICHIERS AUTOEXEC.BAT

Alors que CONFIG.SYS s'occupe de charger les gestionnaires de périphérique nécessaire pour votre système, la tâche d'AUTOEXEC.BAT est de charger automatiquement les programmes d'applications et les modules du système d'exploitation, pour éviter un travail de frappe fastidieux à chaque démarrage.

### **Rappel :**

Le système d'exploitation peut être grossièrement partagé en trois parties :

- Le noyau du système d'exploitation
- Les commandes internes
- Les commandes externes ( programme auxiliaires )

Le noyau du système d'exploitation ( les deux fichiers cachés IO.SYS et MSDOS.SYS, ainsi que le processeur de commande COMMAND.COM, éventuellement DBLSPACE.BIN si votre lecteur est comprimé ) se trouve depuis longtemps dans la mémoire lorsque le fichier AUTOEXEC.BAT est pris en compte. Grâce à COMMAND.COM, les commandes internes sont également disponibles. Seules les commandes externes attendent sur le disque dur d'être appelées.

Certaines des spécifications nécessaires au système ne peuvent pas être effectuées par CONFIG.SYS, car elles ne pourraient plus être modifiées, ce qui n'est pas toujours souhaitable.

Pour exemple : Aux états unis, où le PC à été conçu et où le système d'exploitation a été développé, le clavier des machines à écrire n'était pas le même que le nôtre. Cela tient d'une part au fait que l'anglais ne possède pas de lettres accentuées, mais aussi que les caractères spéciaux sont différents.

On ne pouvait donc pas se servi du clavier QWERTY ( clavier Américain, à cause des six premières lettres ) en France, où l'on utilise le clavier AZERTY. Or le PC standard ne connaît pas cette version du clavier. Il a donc fallu créer différents gestionnaires de clavier en fonction du pays d'origine. On ne pouvait cependant pas charger ces gestionnaires dans le CONFIG.SYS car il fallait changer de définition de clavier à volonté.

C'est pourquoi, ces modules sont chargés, non pas sous forme de gestionnaires de périphérique mais de programme résident par AUTOEXEC.BAT.

### **1°/ PROGRAMMES REDIDENTS ( TSR :Terminate and Stay Resident )**

Un programme normal ( par exemple un traitement de texte ( DOS ) ou un jeu ) prend fin lorsque l'on lui demande et son code disparaît alors de la mémoire.

En se servant de la commande MEM /C /P, on peut le vérifier.

Au contraire, un programme résident prend fin sans être pour autant effacer de la mémoire. On peut l'apercevoir en utilisant la commande MEM qui ne donne pas la même valeur avant et après l'utilisation de ce programme.

On appelle ces programmes, qui demeure en mémoire même après qu'on les quittés, des programmes **TSR** ( Terminate an Stay Resident ).

Ces programmes continuent donc de fonctionner en tâche de fond.

## 2°/ LISTE DES PROGRAMMES TSR DE MSDOS

### ■ APPEND.EXE Recherche les fichiers de données

La variable d'environnement PATH permet de définir un chemin de recherche pour les fichiers de commandes, alors qu'APPEND est destiné à retrouver des fichiers de données - par exemple des textes issus de votre programme de traitement de texte.

*APPEND répertoire ;.... ;répertoire*

### ■ DOSKEY.COM Répétition et macros

DOSKEY.COM sert à créer des Macros, avec lesquelles il est possible de définir des suites complexes de fonctions sous la forme de commandes simples, ce qui permet de gagner du temps.

Exemple : Vous souhaitez créer une macro, appelée COPIE, destinée à copier tous les fichiers qui ont pour extensions TXT sur la disquette. On écrira

*DOSKEY COPIE=COPY \*.TXT A :*

### ■ FASTOPEN.EXE Conserver l'emplacement des fichiers

Cette commande met en place une sorte de mémoire miniature. Elle conserve les répertoires pour le nombre indiqué de fichiers, afin de les retrouver sur le support de données

Cette commande est conseillée que si vous ne n'avez pas installé de mémoire cache ( SMARTDRV.EXE )

*FASTOPEN C :=50 D :=20*

### ■ GRAPHICS.COM Impression d'un écran

Si vous voulez sortie sur imprimante une copie de l'écran avec la touche <Impr>, ce n'est possible qu'avec les écrans de texte. Si vous voulez copier en revanche un écran graphique, vous devez charger le programme résident GRAPHICS.

*GRAPHICS*

### ■ INTERSVR.EXE Lecteur en série

Pour relier deux ordinateurs par l'intermédiaire de l'interface série.

*INTERSVR lect : {option}*

### ■ KEYB.COM Gestionnaire de clavier

Vous placerez cette commande dans AUTOEXEC.BAT, pour avoir à votre disposition le clavier souhaité dès que vous allumerez l'appareil.

Quand vous intégrez KEYB.COM, le pays voulu doit être indiqué à l'aide d'une abréviation et avec mention du fichier contenant les données, donc du gestionnaire KEYBOARD.SYS :

*KEYB FR,,C :\DOS\KEYBOARD.SYS*

Entre les deux virgules, il est possible d'indiquer une table de police de caractères.

## ■ **MODE.COM**

### Configuration générale

MODE est une commande universelle pour la configuration. Elle n'est toutefois utilisée que très rarement, pour modifier la taille des caractères à l'écran. Voir manuel d'utilisation MSDOS.

## ■ **MOUSE.COM**

### Gestionnaire souris

Le gestionnaire de souris MOUSE.COM est fourni avec MS DOS, mais ne fonctionne qu'avec les souris Microsoft ou une souris compatible.

Il existe deux façons de charger un gestionnaire de souris. D'une part comme gestionnaire de périphérique MOUSE.SYS par l'intermédiaire de CONFIG.SYS, d'autre part comme programme résident ( TSR ) dans AUTOEXEC.BAT.

Les deux ne doivent pas être utilisés en même temps.

## ■ **MSCDEX.EXE**

### Lecteur de CDROM

la commande MSCDEX sert à intégrer un lecteur de CDROM dans le système de lecteur, qui doit avoir été annoncé auparavant avec un gestionnaire de périphérique dans CONFIG.SYS. ( Gestionnaire fourni sur disquette avec le lecteur CD ROM ).

Voir manuel fourni avec le lecteur CDROM pour chargé le gestionnaire et le programme MSCDEX.EXE.

## ■ **PRINT.EXE**

### Impression en tâche de fond

Si vous ne voulez pas imprimer à partir de votre programme, la commande PRINT vous permet d'imprimer plusieurs fichiers d'impression l'un à la suite de l'autre pendant que vous travaillez à autre chose sur votre ordinateur. PRINT est donc ce qu'on appelle un SPOOLER d'impression, qui permet d'imprimer en tâche de fond.

PRINT dispose de réglage possible : Voir manuel MSDOS ou HELP

Pour installer normalement PRINT, utilisez la commande suivante :

*PRINT*

Une fois l'installation réalisée, la commande :

*PRINT nom\_fichier*

place le fichier dans la file d'attente.

## ■ **SHARE.EXE**

### Accès aux fichiers

Si vous travaillez dans un réseau ou si vous voulez utiliser un programme de façon multiple sous Windows, vous devez charger SHARE, afin de régler l'accès aux fichiers,, c'est à dire des accès simultanés au même fichiers. En dehors de cette circonstance, vous n'avez pas besoin de cette commande.

*SHARE*

## ■ SMARTDRV.EXE

### Accès au disque dur

Quand un ordinateur dispose d'un processeur i386 ou i486, les opérations contrôlées par le processeur se déroulent si rapidement que le disque dur ne peut pas suivre et devient en quelque sorte le goulot d'étranglement du système. Pour accélérer les accès au disque dur, on installe dans la mémoire principale un MEMOIRE CACHE, qui sert de mémoire intermédiaire en vue de recevoir les données en provenance du disque dur. Quand vous accéder au disque de cette façon, les données sont d'abord chargées dans cette mémoire cache rapide pour être traitées à partir de là. Elles ne sont appelées directement à partir du disque qu'en cas d'accès erroné.

La mémoire cache est installée et gérée à l'aide de SMARTDRV.EXE. Deux paramètres sont très importants dans ce contexte : d'une part la taille du cache et d'autre part la taille minimale du cache avec WINDOWS.

Dans AUTOEXEC.BAT, la commande d'appel se présentera donc ainsi :

*SMARTDRV 2048 1024*

Cette commande crée tout d'abord une mémoire cache de 2 Mo, mais donne à Windows le droit de la réduire à 1024 soit 1 Mo.

## ■ UNDELETE.EXE

### Fichiers supprimés

UNDELETE est un programme destiné à restaurer des fichiers supprimés. Ce programme se lance sous DOS avec la commande UNDELETE seulement ainsi qu'avec différents paramètres.

La méthode la plus sûre est surveillance des suppressions, qui conserve les fichiers supprimés un certain nombre de jours dans un répertoire caché. UNDELETE est alors chargé sous forme de programme résident et surveille les suppressions.

Si vous voulez installer une surveillance des suppressions de ce type, vous devez appeler UNDELETE.EXE avec l'option /LOAD :

*UNDELETE /LOAD*

Le programme cherche alors un fichier nommé UNDELETE.INI, pour charger en mémoire les paramètres qui y sont déposés et y enregistrer avec soin ses activités en rapport avec les suppressions.

## ■ VSAFE.COM

### Défense anti-virus

VSAFE est un programme résident en mémoire principale, dont la tâche est de surveiller les activités suspectes qui pourraient signaler la présence d'un virus. VSAFE est un anti-virus mais de faible capacité, car les anti-virus doivent être continuellement remis à jour en fonction des nombreux virus qui sont découverts. Il ne faut pas trop se fier à lui, et acheter un anti-virus sérieux.

### 3°/ MEMOIRE D'ENVIRONNEMENT

Le fichier AUTOEXEC.BAT a donc pour tâche, en premier lieu de charger les programmes résident ( TSR ) du système d'exploitation que l'utilisateur juge nécessaire dans son environnement de travail.

D'autre part, ce fichier BATCH exécuté automatiquement a aussi pour tâche de charger dès le démarrage du système les programmes avec lesquels l'utilisateur veut travailler.

Pourtant, ce n'est encore là qu'une partie de la réalité. En effet, le fichier AUTOEXEC.BAT est chargé d'une autre tâche : La définition des contenus dans la mémoire d'environnement.

La mémoire d'environnement est nommée ainsi car elle constitue un secteur de mémoire se trouvant dans voisinage immédiat du processeur de commandes ( COMMAND.COM ), pour que ce dernier puisse accéder aussi rapidement que possible à ses données. C'est là que sont déposées toutes les données nécessitées par COMMAND.COM pendant qu'il travaille. Il s'agit donc d'une sorte de dépôt, où d'autres programmes placent aussi leurs données, pour que le processeur de commandes les ait sous la main.

Les données déposées ici sont nommées **variable d'environnement**, car leur contenu peut varier du tout au tout. Le principe est très simple : La variable reçoit un nom et un contenu qui lui est affecté avec un signe d'égalité.

#### a) Définition d'une variable :

Il existe deux moyens de définir ou de modifier une variable d'environnement :

- Les deux variables du chemin de recherche PATH et du caractère d'invite spécifié avec PROMPT sont définies à l'aide d'une commande normale, par exemple

*PATH C : \DOS*

- Toutes les autres variables sont définies à l'aide de la commande SET. La syntaxe de cette commande est :

*SET variable={valeur}*

où l'indication d'une valeur est en réalité optionnelle. Si vous définissez une variable nommée TEMP par :

*SET TEMP=C : \TEMP*

Vous lui affectez la valeur C : \TEMP et vous indiquez ainsi que les fichiers temporaires doivent être déposés dans ce répertoire.

L'appel de

*SET*

sans paramètre affiche le contenu actuel de la mémoire d'environnement, donc l'ensemble des variables qui figurent dans la mémoire, accompagnées de leurs valeurs.

#### b) Taille maximale de la mémoire d'environnement :

La mémoire d'environnement ne comprend que 256 caractère dans sa définition standard, ce qui est le plus souvent très insuffisant, car le chemin de recherche à lui tout seul peut être très volumineux

Dans un cas pareil, le mieux est d'utiliser la commande SHELL, prévue pour intégrer un processeur de commande étranger.

Cette commande permet de réserver plus de mémoire d'environnement, qui ne sera pas mise toutefois à la disposition de la mémoire de travail.

Ainsi la commande

```
SHELL C:\COMMAND.COM C:\ /E:1024 /P
```

charge le processeur de commandes mais affecte de manière permanente une mémoire d'environnement de 1024 octets pour les besoins de ce processeur de commandes.

### **c) Variable standard du système d'exploitation :**

#### ■ **BREAK**                      Utilisation du <Ctrl>+<C>

Si break est défini sur ON, et dans ce cas seulement, le processeur de commandes est constamment aux aguets pour voir si vous avez interrompu un accès aux lecteurs avec Ctrl C. Vous pouvez le demander à partir de CONFIG.SYS. Vous pouvez aussi définir dans le fichier AUTOEXEC.BAT une variable d'environnement nommée BREAK et lui donner la valeur ON. L'effet sera exactement le même.

```
BREAK ON
```

#### ■ **COMSPEC**                      Chemin du processeur de commande

Sous le nom de variable est déposé le nom de processeur de commande chargé, ainsi que l'indication précise du chemin d'accès au processeur. Puisque le processeur de commandes est parfois retiré partiellement de la mémoire, il faut que le chemin soit connu pour qu'il puisse être chargé rapidement en cas de besoin. Cette variable est créée automatiquement au moment où l'on charge le processeur de commandes, et vous n'avez pas besoin d'intervenir.

#### ■ **CONFIG**                      Menu de démarrage

Cette variable est définie quand vous créez un menu de démarrage dans le fichier CONFIG.SYS pour pouvoir réaliser des embranchements vers des actions diversifiées dans AUTOEXEC.BAT en relation avec la commande sélectionnée. Avec :

```
GOTO %CONFIG%
```

vous effectuez ainsi un saut à la marque spécifiée.

#### ■ **DIRCMD**                      Exécution de DIR

La commande DIR sert à afficher le catalogue d'un support de données. Si vous voulez définir de façon générale le mode d'exécution de cette commande( tri des fichiers, affichage page par page, etc... ), vous pouvez déposer la variable DIRCMD dans la mémoire d'environnement, en lui donnant pour valeur les options correspondantes. Ainsi la commande suivante :

```
SET DIRCMD= /P /O :E
```

définira à chaque fois que vous taperez la commande DIR, exécutera cette commande en réalisant un affichage page par page ( /P ) et en réalisant un tri en fonction des extensions ( /O:E ).

## ■ MSDOSDATA

### Dépôt d'informations DOS

Les programmes MSBAXKUP, MWBACKUP, VSAFE, MWAV, et UUNDELETE déposent des informations utiles dans des fichiers créés normalement dans le répertoire racine. Pour ne pas encombrer la racine, on peut les déposer dans un autre répertoire :

```
SET MSDOSDATA=C:\DOS\DONNEES
```

Pour déposer ces fichiers dans un autre répertoire, vous indiquerez le chemin d'accès de ce dernier, avec la variable MSDOSDATA, en sachant toutefois que le répertoire doit être créé d'abord.

## ■ PATH

### Chemin de recherche

Si vous déposez vos programmes dans des répertoires distincts, cette distribution présente un inconvénient de taille à côté de ses nombreux avantages : Vous êtes obligé d'indiquer à chaque fois, au moment de lancer un logiciel, le répertoire dans lequel il se trouve.

Exemple :      Ainsi, au lieu de taper simplement :      *WORD*  
                  Vous êtes obligé d'écrire :            *\WORD55\WORD*

Pour simplifier cet appel, vous pouvez indiquer dans la variable d'environnement ce qu'on appelle un chemin de recherche, soit une suite de répertoires dans lesquels le processeur de commande doit effectuer la recherche pour retrouver un programme donné. Il parcourt alors les répertoires placés dans la variable jusqu'à ce qu'il ait trouvé celui-ci et il le lance.

Les répertoires doivent être simplement inscrits l'un à la suite de l'autre, séparés par des points virgules :

```
PATH C:\DOS ;C:\DOS ;C:\WINDOWS
```

Le chemin de recherche ne doit comprendre plus de 127 caractères.

## ■ PROMPT

### Invite

Cette commande permet de définir l'aspect du caractère d'invite MS DOS, le prompt du système d'exploitation. Par défaut, on a :

```
PROMPT $P$G
```

Pour rajouter l'heure, on aura :

```
PROMPT $T $P$G
```

## ■ TEMP

### Fichiers temporaires

Il est parfois inévitable, pendant l'exécution d'un programme, de créer la copie d'un fichier, copie qui sera supprimée ultérieurement. C'est le principe des fichiers temporaires, c'est à dire des fichiers qui n'interviennent que provisoirement.

Exemple : Les traitements de texte

Il est certes prévu que ces fichiers soient effacés quand ils ne sont plus nécessaires, mais il arrive souvent qu'ils ne le soient pas, en particulier quand un programme se bloque et vous oblige à un redémarrage. Dans ce cas, ces fichiers restent sur le disque et encombrant l'espace de leur présence inutile.

Avec la variable d'environnement TEMP, vous demandez à tous les programmes de créer leurs fichiers

temporaires uniquement dans le répertoire spécifié ici. De cette façon, vous savez où se trouvent les fichiers temporaires qui restent éventuellement sur le disque et vous pouvez les effacer rapidement.

*SET TEMP :C :\TEMP*

## ■ TMP

Fichiers temporaires

Tout ce qui vaut pour la variable TEMP vaut aussi pour cette variable, à ceci près que certains programmes, en particulier les logiciels américains, consultent le tableau d'affichage à la recherche d'une variable TMP. Vous serez donc avisé de définir également cette variable, car vous ne pouvez pas savoir à l'avance quelle est la variable utilisée par tel ou tel logiciel.

## ■ WINPMT

Prompt de WINDOWS DOS

Si vous êtes obligé de revenir provisoirement de WINDOWS au système d'exploitation DOS, vous pouvez le faire par l'intermédiaire de la boîte DOS dans le groupe principal ( WINDOWS 3.1 ). Vous êtes en cession MS DOS et vous revenez à WINDOWS par EXIT. Vous pouvez grâce à cette commande, définir un prompt spécial pour cette session DOS.

*SET WINPMT= Nous somme dans Windows ! \$P\$G*

### COMMANDES SUPPLEMENTAIRES

D'autres commandes sont du CONFIG.SYS et de l'AUTOEXEC.BAT existent, nous en étudierons plus tard, les commandes pour réaliser des menus de démarrage.

## 4°/ IGNORER CONFIG.SYS ET AUTOEXEC.BAT AU LANCEMENT

Depuis la version 6 de MS DOS, vous pouvez demander ( pour de multiples raisons ) un démarrage système ignorant les fichiers de configurations avec la touche <F5>, pendant que le message

### *Démarrage de MS-DOS...*

est à l'écran ( et seulement alors !! ). Ce démarrage pur et simple ignore les fichiers de démarrage CONFIG.SYS et AUTOEXEC.BAT.

En appuyant sur la touche <F8>, vous réaliserez par contre un démarrage sélectif, en répondant O ou N derrière chaque ligne de CONFIG.SYS et ensuite AUTOEXEC.BAT. Vous pourrez grâce à cette commande exclure certaines lignes du CONFIG.SYS et de l'AUTOEXEC.BAT.

Ce procédé est très pratique lorsque vous avez des erreurs dans les fichiers de configurations CONFIG.SYS et AUTOEXEC.BAT.

## REALISATION DU CONFIG.SYS

Dans la description qui suit, nous partons pour notre exemple d'un répertoire C:\DOS, dans lequel se trouve les gestionnaires et les programmes.

Si vous utilisez un disque dur qui a besoin d'un gestionnaire spécial afin de pouvoir être adressé, celui-ci doit être chargé avant tout les autres, car les gestionnaires qui viennent ensuite ne peuvent être cherchés et trouvés que si le disque est d'abord ouvert.

### MODELE DE BASE DE CONFIG.SYS

```
DEVICE=C :\DOS\HIMEM.SYS
```

Installation de la mémoire étendue

```
DEVICE=C :\DOS\EMM386.EXE NOEMS
```

Accès à la mémoire supérieure sans la mémoire EMS

```
DOS=HIGH, UMB
```

Charger le système d'exploitation en HMA et UMB

```
DEVICEHIGH=C :\DOS\SETVER.SYS
```

Simulation de DOS

```
COUNTRY=033,,C :\DOS\COUNTRY.SYS
```

Définir les spécifications des données nationales

```
SHELL=C :\COMMAND.COM C :\DOS /E :1024 /P
```

Charger le processeurs de commande avec 1024Ko de mémoire d'environnement

```
BUFFERS=20
```

Création de 20 tampons accès

```
FILES=20
```

20 fichiers max peuvent être ouvert simultanément

Bien sûr, il ne peut y avoir de modèle valide de manière générale, puisque c'est précisément l'une des caractéristiques de la configuration par l'intermédiaire de CONFIG.SYS et AUTOEXEC.BAT : Elle adapte l'ordinateur à l'environnement de travail de l'utilisateur et les composants, à une coopération pacifique.

- Ce fichier est toujours valable. C'est la configuration de base pouvant fonctionner tout le temps.
- Il faudra ensuite adapter votre CONFIG.SYS à l'ordinateur en question en rajoutant les lignes nécessaires.

### Remarques complémentaires :

- Si vous voulez utiliser la mémoire supérieure dans le CONFIG.SYS, installé au préalable par EMM386.EXE, vous devez modifier pour tous les gestionnaires qui peuvent l'être ( voir cour : Gestionnaire avec \* ) la commande de chargement DEVICE par DEVICEHIGH.

Exemple :     DEVICEHIGH :C :\DOS\ANSI.SYS

- Pensez toujours qu'en cas de problème, vous avez le programme MEMMAKER qui vous dépannera.

## REALISATION DE AUTOEXEC.BAT

Les tâches de AUTOEXEC.BAT ont déjà été mentionné précédemment :

- Lancement de modules résidents du système d'exploitation pour la configuration de l'ordinateur et de votre environnement de travail, ainsi que pour le chargement des programmes externes, comme le gestionnaire d'écran de votre carte graphique.
- Définition des variables d'environnement du système d'exploitation, ainsi que des variables qui seront recherchées par les programmes d'application dans la mémoire d'environnement.
- Lancement des programmes comme WINDOWS ou votre traitement de texte au démarrage de l'ordinateur.

### RAPPELS :

Le fichier AUTOEXEC.BAT est beaucoup plus libre que le CONFIG.SYS. Celui-ci ne peut être effectif qu'après un redémarrage du système alors que AUTOEXEC.BAT peut-être modifié après coup.

### PROGRAMME RESIDENT :

Les programmes résidents du systèmes d'exploitation sont pour l'essentiel les suivant :

- Mémoire cache SMARTDRV.EXE, et définition de sa taille
- Gestionnaire de clavier souhaité ( Français ou autre ) avec KEYB.COM
- Surveillance des fichiers supprimés avec UNDELETE.EXE
- SHARE, si vous voulez faire fonctionner sous Windows plusieurs instances d'un programme indépendamment l'une de l'autre.

### BLOCS DE MEMOIRE SUPERIEURE ( UMB )

Les blocs de mémoire supérieure, dont nous avons déjà parlé, se trouvent dans le segment d'adaptation entre 640 Ko et 1Mo. Ils peuvent être utilisés pour recevoir des gestionnaires de périphérique ( CONFIG.SYS ) ou des programmes résidents ( AUTOEXEC.BAT ).

- Pour charger un programme résident dans la mémoire supérieure, utiliser la commande **LOADHIGH** que vous pouvez aussi introduire de manière abrégée sous la forme :

**LH**

### PROMPT :

\$D	Affiche la date système
\$E	Caractère ESCAPE
\$G	Caractère >
\$H	Supprime le caractère précédent
\$L	Caractère <
\$N	Lecteur actuel, prompt standard avec \$G
\$P	Répertoire actuel
\$T	Affiche l'heure système

Tous les caractères qui ne sont pas introduits par un signe dollar ( \$ ) sont reportés tel quels dans le prompt.

Ainsi : *PROMPT Aujourd'hui : \$D - \$T \$P\$G*

on aura à l'écran le prompt :

*Aujourd'hui : 14 - 6 - 1993 12 :34 :58 C :\DOS>*

### MODELE DE AUTOEXEC.BAT

@ECHO OFF

SET TEMP=C :\TEMP

SET TMP=C :\TEMP

IF EXIST C :\TEMP\\*.\* DEL C :\TEMP\\*.\*

PATH C :\DOS ;C :\WINDOWS ;CD :\WORD

PROMPT \$P\$G

LH SMARTDRV.EXE

LH KEYB FR,,C :\DOS\KEYBOARD.SYS

LH MOUSE.COM

WIN

Désactive l'affichage à l'écran toutes les commandes qui suivent.

Création d'un variable d'environnement TEMP et TMP placé dans le répertoire TEMP et qui effaceront automatiquement tous les fichiers temporaires restés dans ce répertoire.

Création de chemin d'accès

définition du prompt

Création de mémoire cache

Définition du clavier

Gestionnaire souris

Lancement automatique de Windows

## OPTIMISER LA CONFIGURATION AVEC MEMMAKER.EXE

Nous allons décrire maintenant le travail à effectuer avec ce programme auxiliaire.

Ce programme quoique très efficace, il faut cependant avoir quelques notions sur la configuration mémoire pour pouvoir l'utiliser efficacement.

Le programme MEMMAKER doit être lancé sous MS DOS, vous devez quitter tous les logiciels lancés, en particulier WINDOWS.

Pour lancer le programme, il suffit de taper à l'invite MS DOS la commande :

*MEMMAKER*

Une fois lancé, le programme vous donnera des informations sur lui-même et sur ses principes de fonctionnement.

Avec F3, vous pouvez à tout moment interrompre le processus, en annulant les modifications éventuelles.

Pour être en sécurité complète, nous vous recommandons de réaliser des copies des fichiers CONFIG.SYS et AUTOEXEC.BAT ainsi que WIN.INI s'il existe ( Fichier WINDOWS ). On nomme souvent ces copies avec les extensions OLD ou BAK.

Il y a deux modes d'exécution du programme :

- MODE EXPRESS
- MODE PERSONNALISE

Le premier est recommandé si vous êtes débutant, car les risques de commettre des erreurs sont minimes.

Le second est recommandé pour des personnes plus expérimentée.

### MODE EXPRESS :

Le déroulement des opérations en mode express est très simple :

Le programme vous demande si vous voulez utiliser de la mémoire paginée ( EMS ). Si vous répondez **NON** à cette question ( réponse par défaut ), le gestionnaire EMM386.EXE placé dans CONFIG.SYS sera utilisé uniquement pour déposer le système d'exploitation dans les blocs de mémoires supérieure ( UMB ). Si vous répondez **OUI**, le programme crée la mémoire paginée.

C'est tout ce que vous avez à faire. En cas de doute, appelez l'aide avec **F1**.

Lorsque vous avez répondu également à cette question, il ne vous reste plus qu'à enlever les disquettes qui se trouvent dans les lecteurs, car MEMMAKER fait redémarrer l'ordinateur, pour tester les divers réglages en vigueur dans l'appareil. Ces opérations peuvent même se renouveler plusieurs fois.

- Si un programme est lancé à partir de votre AUTOEXEC.BAT, par exemple WINDOWS, quittez-le pour que MEMMAKER puisse continuer son travail.
- S'il devait surgir un problème, éteignez l'ordinateur et rallumez-le au bout de 10 secondes. MEMMAKER restaure alors l'état initial.

Au bout d'un certain temps, le programme vous demande si l'appareil a démarré correctement, car c'est maintenant la configuration proposée qui préside au processus.

Si aucun problème n'est apparu, un écran vous informe enfin sur la mémoire de travail disponible à partir de maintenant, et donc sur le bénéfice que vous avez réalisé grâce à MEMMAKER.

- Les anciens fichiers de démarrage CONFIG.SYS et AUTOEXEC.BAT, ainsi que WIN.INI s'il existe, reçoivent l'extension UMB. Ils sont placés dans le répertoire DOS.

## **MODE PERSONNALISE**

Il faut pour cela savoir répondre à différentes questions :

### **■ Spécifier les gestionnaires TSR à inclure ?**

Par défaut, MEMMAKER prend en compte tous les gestionnaires de périphérique et programme résident dans le processus d'optimisation. Dans certains cas, il se peut qu'un logiciel pose des problèmes. Vous le remarquez au fait qu'il ne démarre pas ou qu'il provoque le blocage de l'ordinateur. Vous pouvez éviter ces problèmes en excluant dès l'abord ce logiciel du processus d'optimisation.

### **■ Examiner la mémoire supérieure de manière détaillée ?**

Par défaut, MEMMAKER indique à EMM386 qu'il doit chercher des blocks de mémoire supérieure libre aux adresses F000 à F7FF. Quand on place des blocs de mémoire supérieure à ces adresses, il arrive que certains ordinateurs ne démarrent pas correctement. Si vous répondez non, les blocs de mémoire supérieure ne seront cherchés qu'aux adresses C600 à EFFF et tous conflits seront de la sorte évités.

### **■ Optimiser la mémoire supérieure pour l'utilisation de Windows ?**

par défaut, le système est optimisé de telle sorte que les logiciels MS DOS aient le plus possible de mémoire conventionnelle à leur disposition quand ils fonctionnent sous Windows. Si vous n'utilisez pas Windows, répondez NON à cette question.

### **■ Libérer la région monochrome ( B000-B7FF) pour les programmes ?**

Par défaut, 32 Ko de mémoire haute ( Adresse B000-B7FF ) sont réservés pour une carte graphique monochrome. Ce secteur de mémoire reste inutilisé sur la plupart des ordinateurs équipés de carte EGA et VGA.

**Avec une carte SVGA, vous devez répondre NON.**

### **■ Garder les exclusions et inclusions actuelles de EMM386 ?**

Normalement, les paramètres en vigueur avec EMM386.EXE sont pris en compte, puisqu'ils n'ont pas été indiqués au hasard. Si vous répondez NON, MEMMAKER peut utiliser ses propres paramètres.

## ■ Déplacer la zone de données étendue BIOS de la mémoire ?

Le secteur BIOS étendu ( Extended BIOS Data Area,... EBDA ) se trouve normalement dans la mémoire conventionnelle. Par défaut, MEMMAKER déplace ce secteur EBDA dans la mémoire supérieure, ce qui permet à nouveau de gagner quelques octets supplémentaires pour les applications.

Une fois répondu à toutes ses questions, le processus est le même que pour le MODE EXPRESS.

## ANNULATION DES CONFIGURATIONS

Vous avez beau avoir tout prévu, il peut toujours se produire des choses inattendues, et il peut arriver en particulier qu'une configuration ne fonctionne pas exactement comme prévu.

Vous pouvez annuler toutes les modifications que MEMMAKER avec la commande :

*MEMMAKER /UNDO*

Ce programme fonctionne avec les fichiers UMB du répertoire DOS copiés avant l'opération.

Le plus simple cependant serait de remplacer les fichiers CONFIG.SYS, AUTOEXEC.BAT et WIN.INI ( si existant ), par les anciens qui ont pour extensions UMB dans le répertoire DOS.

## MENU DE DEMARRAGE

Une nouveauté très utile de la version 6.0 est la possibilité de réaliser le démarrage de l'ordinateur à partir d'un menu et ainsi de pouvoir travailler avec des configurations différentes.

Vous pouvez faire en sorte de travailler parfois avec de la mémoire paginée et parfois sans.

Avant la version 6.0, on devait créer différents fichiers pour ces configurations. Ces fichiers étaient baptisés chacun à leur tour CONFIG.SYS et AUTOEXEC.BAT, à chaque fois qu'on les faisait intervenir au démarrage. On devait donc, à chaque fois, rebooter l'ordinateur.

Les opérations sont beaucoup plus simples depuis qu'il existe la version 6.0 ; car il est maintenant possible de créer un menu de démarrage, qui apparaît à chaque fois que vous allumez l'appareil et qui laisse à l'utilisateur, au moyen d'un menu, le choix entre diverses configurations pour le démarrage.

### I/ CONFIG.SYS

Voici un exemple :

```
[MENU]
MENUITEM=SANS_EMS, Sans mémoire paginée
MENUITEM=AVEC_EMS, Avec mémoire paginée

[SANS_EMS]
REM On choisit ici le CONFIG.SYS pour le démarrage sans mémoire paginée
...

[AVEC_EMS]
REM On choisit ici le CONFIG.SYS pour le démarrage avec mémoire paginée
...
```

Il n'est pas possible de faire plus simple. LA commande MENUITEM dans la section [MENU] permettent de définir des entrées dans le menu ( Au maximum 9 ), qui apparaissent ultérieurement elles-mêmes comme des sections contenant les listes des commandes CONFIG.SYS

Si l'ordinateur est lancé avec le CONFIG.SYS écrit avec cet exemple, vous verrez apparaître au démarrage le menu suivant :

#### Menu de démarrage pour MS - DOS 6

1. Sans mémoire paginée
2. Avec mémoire paginée

*Entrez votre choix : 1*

Dans la ligne du bas, un message vous rappelle que vous pouvez exécuter le démarrage de manière sélective avec F5 et F8. ( Voir cour )

La première entrée du menu est placée en surbrillance. Vous pouvez déplacer la surbrillance à votre gré avec les touches de directions, où avec le numéro du choix.

Quand vous appuyer sur ENTREE, le CONFIG correspondant est lancé.

#### a) Valeur par défaut

Avec la commande MENUDEFAULT, vous définirez une valeur par défaut, si celle-ci n'est pas le point 1 du Menu.

Vous pourrez indiquer également au bout de combien de temps d'inaction cette valeur sera choisie automatiquement.

```
[MENU]
MENUITEM=SANS_EMS, Sans mémoire paginée
MENUITEM=AVEC_EMS, Avec mémoire paginée
MENUDEFAULT= AVEC_EMS, 10
```

```
[SANS_EMS]
REM On choisit ici le CONFIG.SYS pour le démarrage sans mémoire paginée
...
```

```
[AVEC_EMS]
REM On choisit ici le CONFIG.SYS pour le démarrage avec mémoire paginée
...
```

La valeur par défaut dans ce menu de démarrage simple est le numéro 2. Au bout de 10 secondes, l'ordinateur démarre avec cette configuration.

#### b) Couleur de menu

La commande MENUCOLOR permet de définir les couleurs pour le texte du menu et pour l'arrière plan.

Numéro	Couleur	Numéro	Couleur
0	Noir	8	Gris
1	Bleu	9	Bleu clair
2	Vert	10	Vert clair
3	Cyan	11	Cyan clair
4	Rouge	12	Rouge clair
5	magenta	13	magenta clair
6	marron	14	jaune
7	gris clair	15	blanc

Exemple : Menucolor 4,7  
Vous obtiendrais des lettres rouges sur fond gris clair.

#### c) Sous menu

Si vous voulez contourner la limitation du nombre d'entrées à 9, ou de créer un sous menu, utilisez la commande SUBMENU :

```
[MENU]
SUB_MENU=SANS_RESEAU,
MENUITEM=AVEC_RESEAU, Avec réseau
MENUDEFAULT= AVEC_RESEAU, 10
MENUCOLOR 2,1
```

```
[SANS_RESEAU]
MENUITEM = PAUL
MENUITEM = JACQUES
```

```
[PAUL]
REM On choisit ici le CONFIG.SYS pour le démarrage sans réseau dans PAUL
```

...

```
[JACQUES]
REM On choisit ici le CONFIG.SYS pour le démarrage sans réseau dans JACQUES
```

...

```
[AVEC_RESEAU]
REM On choisit ici le CONFIG.SYS pour le démarrage avec réseau
```

...

Le menu par défaut sera le menu avec réseau au bout de 10 secondes. On aura du texte vert sur fond noir.

#### d) Définir des blocs

Il arrive souvent que vous ayez défini un bloc de commandes dont vous avez besoin quelques lignes plus loin. Pour ne pas être obligé de tout recopier, vous pouvez dans ce cas donner un nom à ces lignes.

```
[MENU]
MENUITEM= NORMAL, Configuration normale
MENUITEM= EMS, Mémoire paginée
```

```
[NORMAL]
DEVICE=C :\DOS\HIMEM.SYS
DOS=HIGH, UMB
```

```
[EMS]
INCLUDE=NORMAL
DEVICE=C :\DOS\EMM386.EXE
```

Tout le contenu du bloc NORMAL est ainsi inséré avec la commande INCLUDE.

#### e) Définir des blocs généraux

Si vous voulez utiliser des lignes de commande valides dans tous les cas possible, utilisez la commande COMMON.

```
[MENU]
MENUITEM=SANS_EMS, Sans mémoire paginée
MENUITEM=AVEC_EMS, Avec mémoire paginée
```

```
[SANS_EMS]
REM On choisit ici le CONFIG.SYS pour le démarrage sans mémoire paginée
...
```

```
[AVEC_EMS]
REM On choisit ici le CONFIG.SYS pour le démarrage avec mémoire paginée
...
```

```
[COMMON]
REM Vous placerez ici les commandes valides pour tous.
...
```

## II/ AUTOEXEC.BAT

On peut réaliser le même travail avec AUTOEXEC.BAT.

Supposons vous vouliez d'une part charger Windows dès le démarrage et d'autre part travailler tout de suite à partir du DOS.

Vous définissez toutes les spécifications communes tels que HIMEM.SYS, ect..  
Passez ensuite à une marque avec

```
GOTO %CONFIG%
```

Cette commande permet de joindre le CONFIG.SYS et AUTOEXEC.BAT

Cette marque devra porter exactement le même nom que le bloc correspondant dans le CONFIG.SYS donc :

### CONFIG.SYS

```
[MENU]
MENUITEM=DOS, Démarrage en mode MS DOS
MENUITEM= WIN, Démarrage en mode WINDOWS
```

```
[DOS]
```

```
...
```

```
[WIN]
```

```
...
```

### AUTOEXEC.BAT

```
DEVICE=C:\DOS\HIMEM.SYS
```

```
GOTO %CONFIG%
```

```
:DOS
```

```
...
```

```
:WIN
```

```
...
```