

La mémoire sous DOS

Les microprocesseurs ne sont pas tous capables de gérer la même quantité de mémoire ; cela dépend du nombre de lignes d'adresses.

- Le 8080 avec un bus d'adressage de 16 lignes pouvait adresser $2^{16} = 64$ ko (cela suffisait à l'époque du CP/M)
- Les 8088 et 8086 des premiers PC avait 20 lignes d'adresses (1 Mo)
- Les 80286 pouvaient adresser jusqu'à 16 Mo (adresse en 24 bits)
- Les 80386 peuvent gérer 4 Go (adresses 32 bits)
- Les Pentium actuels ont 36 bits d'adresse, => 64 Go

La mémoire vue par le DOS

Le MS-DOS a été développé pour les 8088 et 8086. Ces microprocesseurs traitaient des mots de 16 bits et le bus d'adressage en comptait 20. Cela a deux conséquences :

- la segmentation de l'adresse
- l'impossibilité de former des adresses de plus de 20 bits.

1° La segmentation des adresses

Puisqu'il n'y avait pas de registre assez large pour contenir une adresse de 20 bits il a fallu codifier ces adresses au moyen de deux nombres. Elles sont exprimées sous la forme `segment:offset`.

Un segment est ici un bloc mémoire dont l'adresse est un multiple de seize. L'offset exprime un déplacement par rapport à cette adresse de base. La correspondance entre l'adresse physique et l'adresse sous forme `segment / offset` est la suivante :

$$\text{Adresse physique} = \text{Segment} * 16 + \text{Offset}$$

Exemple : L'adresse physique B8000 s'écrit sous la forme B000:8000 mais elle pourrait tout aussi bien s'écrire B800:0000 . Le calcul est relativement simple, à condition bien sûr d'écrire ces nombre en hexadécimal. La multiplication de la valeur du segment par 16 se fait en décalant ce nombre d'un chiffre hexadécimal (4 bits) vers la gauche. On ajoute ensuite la valeur de l'offset.

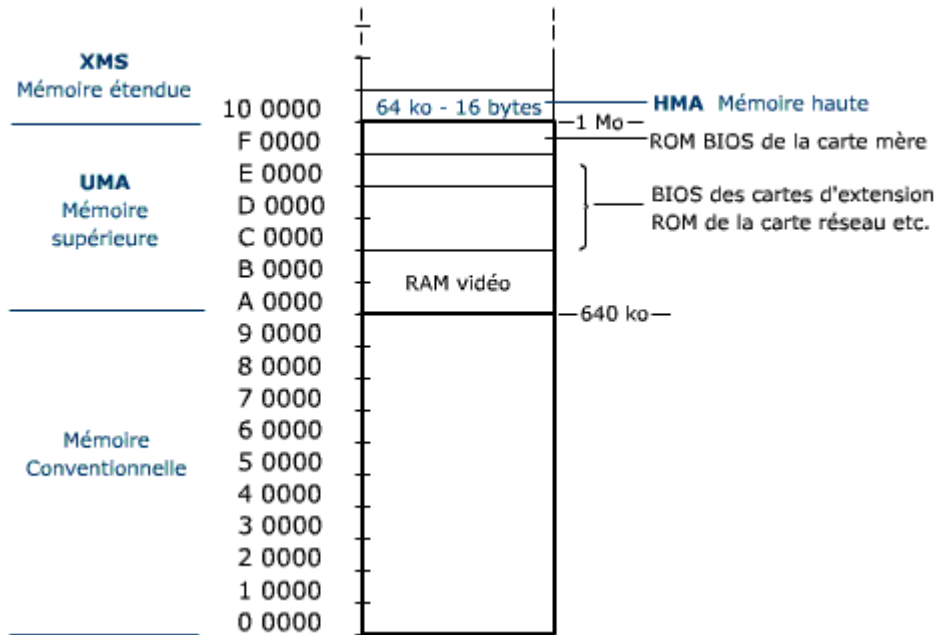
Segment * 16	B000	B800
+ Offset	+ 8000	+ 0000
= Adresse physique	B8000	B8000

° La limite du méga octet

Le DOS n'a pas été prévu pour exploiter plus d'un méga octets (1 Mo).

Seuls les premiers 640 ko pouvaient servir pour y charger les applications, c'est ce qu'on appelait la *mémoire conventionnelle*.

Les derniers 384 ko (la *mémoire supérieure* ou UMA, *Upper Memory Area*) étaient réservés pour la mémoire vidéo, la ROM du BIOS et des plages mémoires à attribuer aux cartes d'extension.



Le DOS occupait de 40 à 90 ko au début de la mémoire. Le reste des 640 ko était occupé par quelques petits programmes résidents et par une application (DOS était monotâche et mono-utilisateur)

N.B. Les applications se contentaient souvent de quelques dizaines de ko. A L'époque, pour exploiter la mémoire restée libre, on avait la possibilité d'y créer un *disque virtuel*. Celui-ci puisqu'il utilisait la mémoire au lieu d'un disque physique avait une vitesse extraordinaire. Les temps ont bien changé puisque actuellement malgré les centaines de méga bytes de mémoire dont nous disposons nous recourons aujourd'hui à de l'espace sur le disque pour y faire un complément de *mémoire virtuelle*.

Mémoire étendue

Les processeur 80286 et suivants disposent de 24 lignes d'adresse ou plus. L'espace mémoire disposé au-delà du premier Mo est appelé *mémoire étendue* (**XMS** *Extended Memory Specification*).

Les systèmes d'exploitation plus récents gèrent la mémoire étendue qu'ils partagent entre les applications. Ces OS sont multitâches, il leur faut un mode de fonctionnement spécial : le **mode protégé**. Le système d'exploitation joue le rôle d'arbitre qui distribue la mémoire aux applications en veillant à ce qu'aucune n'interfère dans les zones mémoires réservées aux autres. Ce mode n'est pas accessible au DOS qui lui continue de fonctionner en **mode réel** comme si le processeur n'était rien de plus qu'un 8086.

Le **mode virtuel 8086** est quant à lui le mode opératoire où l'on se trouve quand on passe en mode invite de commande à partir de Windows. Le système d'exploitation simule l'environnement dont MS-DOS a besoin tout en l'empêchant d'outrepasser les possibilités d'un simple 8086.

Artifices utilisés sous DOS pour disposer de plus de mémoire

Remarque préalable *Les explications qui suivent cherchent uniquement à démystifier les instructions sibyllines que vous risquez de rencontrer dans les fichiers CONFIG.SYS et AUTOEXEC.BAT. Soyons clair, la bonne méthode pour utiliser efficacement la mémoire d'un PC est bien entendu d'utiliser un système d'exploitation récent.*

La mémoire paginée

Ce système développé conjointement par Lotus, Intel et Microsoft offrait une mémoire supplémentaire baptisée **EMS** (*Expanded Memory Specification*) qui ne devait pas nécessairement être contiguë avec le premier Mo. Cet espace de stockage était accessible via un segment de 64 ko à l'adresse D 0000 ou E 0000 de la mémoire haute, accessible au DOS puisqu'elle se trouvait en deçà de la limite du Mo.

Ce segment de 64 ko était subdivisé en quatre fenêtres de 16 ko contenant chacune une copie d'une partie de la mémoire EMS. Pour y accéder, on utilisait un gestionnaire de périphérique appelé **EMM** (*Expanded Memory Manager*- gestionnaire de mémoire paginée) installé au démarrage par la directive telle que `DEVICE = EMM386.SYS` inscrite dans le fichier CONFIG.SYS .

Le gestionnaire de mémoire HIMEM.SYS

Une autre façon d'utiliser cette mémoire étendue sous DOS est de s'en servir comme cache disque. Le disque étant un élément très lent les performances du PC sont nettement améliorées si des accès aux disques peuvent être remplacés par des accès rapides dans des tampons mémoires. C'est pour cette raison que les fichiers AUTOEXEC.BAT sous DOS faisaient systématiquement appel à l'utilitaire SMARTDRV. Le fichier CONFIG.SYS contient quant à lui un appel au gestionnaire de mémoire HIMEM.SYS qui gouverne toute la mémoire étendue et la redistribue aux applications qui se conforment aux spécifications XMS.

Utiliser les emplacements libres dans la mémoire supérieure

Nous avons vu que la mémoire supérieure, 384 Ko entre les adresses A0000 et FFFFF est réservée au matériel : le BIOS de la carte mère, les BIOS et des RAM de cartes d'extension (réseau, vidéo, audio ...). Dans les faits, tout l'espace disponible n'est pas utilisé. Il y a donc des zones disponibles appelées **Upper Memory Blocks UMB**.

D'autre part, quand on compare la mémoire physique installée avec la mémoire reconnue par le système, c'est à dire la mémoire conventionnelle plus la mémoire étendue, on constate qu'il manque bien souvent 384 Ko ! Ces 384 Ko de la mémoire haute ne peuvent cohabiter avec les octets des barrettes RAM qui se trouvent aux mêmes adresses. Ils sont donc ignorés.

Les blocs UMB inutilisés correspondent donc bien à des zones de la RAM qui seraient exploitables. On peut y loger des programmes résidants et des gestionnaires de périphérique. C'est faisable à partir des 386 en installant le driver HIMEM.SYS et le gestionnaire EMM386.EXE puis en indiquant `DOS=UMB` dans CONFIG.SYS

Utiliser la mémoire haute

La mémoire haute **HMA *High Memory Area*** est une zone mémoire qui est accessible en mode protégé bien qu'elle soit au-delà du premier Mo. Il s'agit des 64 ko situés à partir du segment FFFF, autrement dit depuis l'adresse FFFF:0000 jusqu'en FFFF:FFFF.

Cette zone est utilisable en installant le driver HIMEM.SYS. L'instruction `DOS=HIGH` dans CONFIG.SYS commande au DOS de se placer en mémoire haute pour laisser plus de place en mémoire conventionnelle. Il est aussi possible de placer des gestionnaires de périphériques en mémoire haute par les instructions `DEVICEHIGH` dans CONFIG.SYS ou `LOADHIGH` dans AUTOEXEC.BAT.