

www.Mcours.com
Site N°1 des Cours et Exercices Email: contact@mcours.com

MEMOIRE CENTRALE

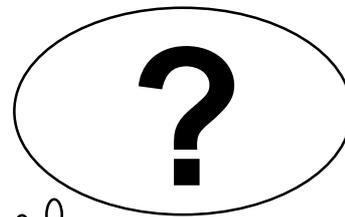


TABLE DES MATIERES

TECHNOLOGIE DES MEMOIRES

☞ Mémorisation informatique	3
☞ Classification et hiérarchie	4
☞ Caractéristiques de la mémoire centrale	6
☞ Les différents types de ROM	7
☞ ROM d'un compatible PC	8
☞ RAM	9
☞ Les différents types de RAM	10
☞ RAM setup d'un compatible PC	12
☞ Installation de RAM sur un PC	13

MEMOIRE DU PC

☞ Limites des compatibles PC	15
☞ Structure générale de la mémoire	16
☞ Mémoire conventionnelle (de 0 à 640 Ko)	17
☞ Résidents (TSR) et drivers	19
☞ Mémoire supérieure (de 640 Ko à 1 Mo)	21
☞ Mémoire paginée EMS	23
☞ Mémoire étendue XMS	24

OPTIMISATION DE LA MEMOIRE CENTRALE

☞ Utilitaires de gestion mémoire	25
☞ Buffer	27
☞ Cache processeur	28
☞ Cache disque	30
☞ Disque virtuel	32

ALLOCATION MEMOIRE

☞ Mémoire virtuelle	33
☞ Segmentation	34
☞ Pagination	36

EXERCICES	38
------------------------	-----------

TECHNOLOGIE : MEMORISATION INFORMATIQUE

MEMOIRES

On appelle **mémoire** tout dispositif capable de stocker des informations et de les restituer à un organe de traitement à n'importe quel moment. Pour cela, une mémoire est constituée des organes suivants :

- ☞ **Cellules de mémoires** qui reçoivent les informations sous forme binaire et les conservent.
- ☞ **Organes d'accès** qui permettent de sélectionner les cellules auxquelles on veut accéder.
- ☞ **Organes de lecture et d'écriture.**

MEMOIRES ADRESSABLES ET NON ADRESSABLES

L'**adressage** désigne la fonction permettant de sélectionner un élément, parmi un ensemble d'éléments de même type, par calcul d'adresse.

Dans les mémoires où les cellules sont parfaitement individualisées, l'emplacement des informations (et par conséquent leur adresse) préexiste à tout rangement de donnée. Ces mémoires sont dites **adressables** (mémoire centrale, disque,...).

Par contre, une mémoire du type bande magnétique n'est pas adressable car il n'y a **pas d'association entre une adresse et l'emplacement de l'information** (les informations sont groupées en blocs et la machine recherche une information en se basant sur le numéro des blocs ou leur contenu).

UNITES DE LA MEMORISATION INFORMATIQUE

☞ **Unité de stockage élémentaire :**

Le **bit** (Binary digIT) est une unité de mémorisation (0 ou 1) valable aussi bien pour la mémoire centrale que pour les disques.

☞ **Unité de mesure de capacité :**

L'**octet** est une unité de codification des caractères sur 8 bits, ayant pour multiples : Ko (Kilo-octet=1024 octets), Mo (Méga-octet=1024 Ko), Go (Giga-octet=1024 Mo), To (Téra-octet=1024 Go).

☞ **Unité d'adressage en mémoire centrale :**

Le **mot** est une unité d'accès mémoire pour le processeur. C'est le plus petit groupement de cellules adressables (pour un disque, l'unité d'adressage est le secteur) ; suivant les ordinateurs, le mot correspond à 1, 2 ou 4 octets (8, 16 ou 32 bits).

☞ **Unités de temps d'accès**

- **ms** (milliseconde) = 10^{-3} s : utilisé pour les mémoires auxiliaires
- **ns** (nanoseconde) = 10^{-9} s : utilisé pour les mémoires centrales (1 million de fois moins)

TECHNOLOGIE : CLASSIFICATION ET HIERARCHIE

CLASSIFICATION DES MEMOIRES

- ☞ **Mémoires statiques** : cellules mémoires pré-positionnées à la construction. Ces mémoires sont généralement adressables et le temps d'accès ne dépend pas de l'adresse. Exemple : **circuits intégrés**.
- ☞ **Mémoires magnétiques cinématiques** : une tête magnétique est mobile par rapport à un support (ou vis-versa). Lorsque ces mémoires sont adressables (disque), le temps d'accès dépend de l'adresse et de l'instant où l'accès est effectué. Ces mémoires sont utilisées comme mémoires auxiliaires (mémoires de masse). Exemples : **disques, bandes**.
- ☞ **Mémoires optiques numériques** : les disques optiques offrent une grande capacité, mais sont plus lents que les disques magnétiques. Exemple : **CD-Rom**.

TABLEAUX RECAPITULATIFS

- ☞ **Mémoire centrale** ou primaire (support immobile).

TYPE DE MEMOIRE		VOLATILE	ADRESSABLE	EFFAÇABLE
circuits intégrés	RAM	oui	oui	oui
	ROM	non	oui	non

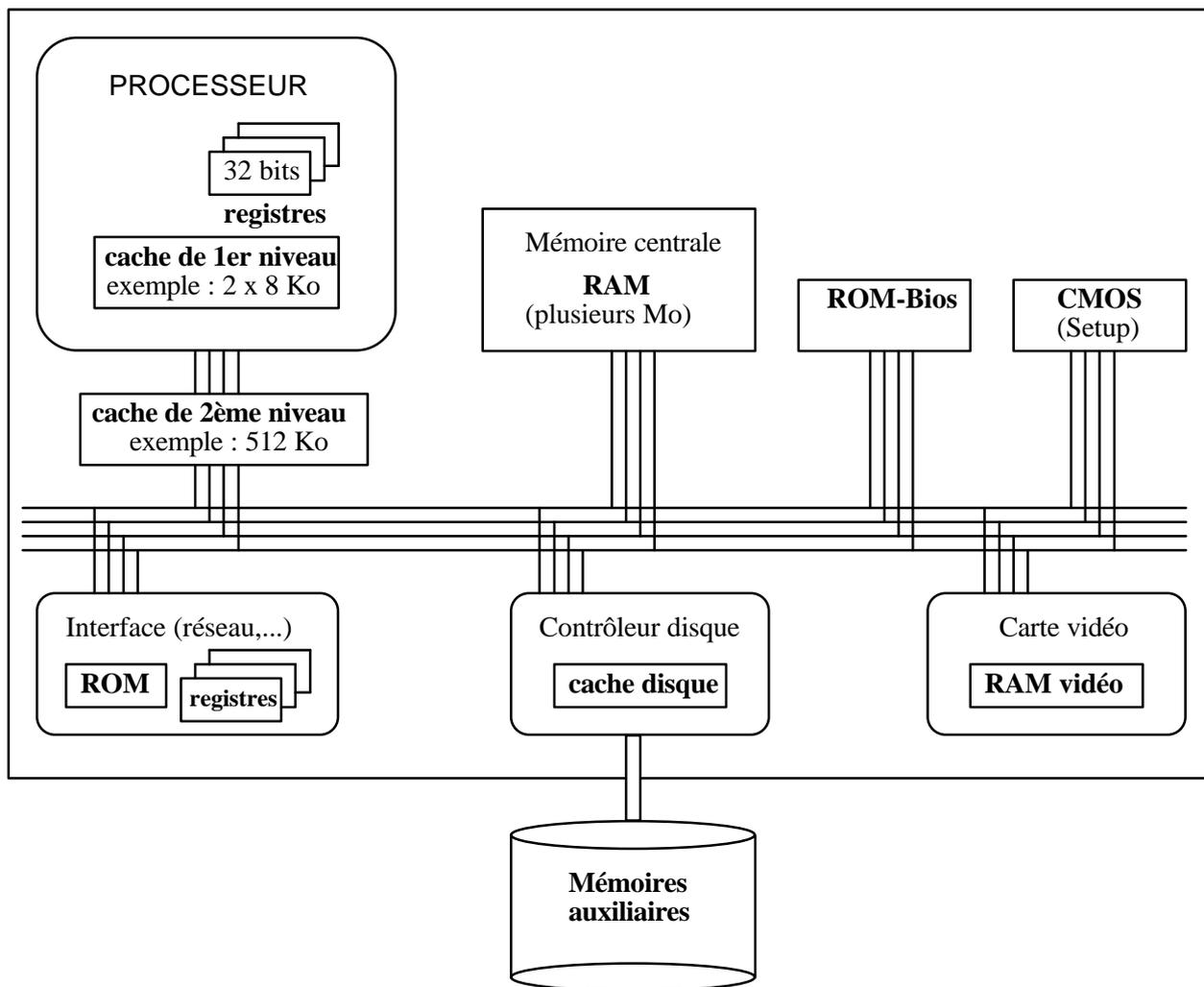
- ☞ **Mémoire de masse** ou secondaire (support en mouvement).

TYPE DE MEMOIRE		VOLATILE	ADRESSABLE	EFFAÇABLE
magnétique	disque et disquette	non	oui	oui
	bande	non	non	oui
optique	CD-Rom	non	oui	non

HIERARCHIE DES MEMOIRES

Dans la hiérarchie des mémoires, les coûts et les temps d'accès sont inversement proportionnels.

- ☞ **Registres** : petites mémoires internes du processeur (ou à une interface).
- ☞ **Caches matériels** : mémoires locales destinées à accélérer les échanges entre deux éléments (autour de 15 ns pour un cache processeur de 2ème niveau).
- ☞ **Mémoire centrale** (de 50 à 70 ns) : espace de travail du processeur.
- ☞ **Mémoire de masse (disque, disquette, bande)** : mémoires magnétiques ou optiques pour lesquelles le temps d'accès est beaucoup plus long que pour la mémoire centrale.



www.Mcours.com
 Site N°1 des Cours et Exercices Email: contact@mcours.com

TECHNOLOGIE : CARACTERISTIQUES DE LA MEMOIRE CENTRALE

MEMOIRE CENTRALE

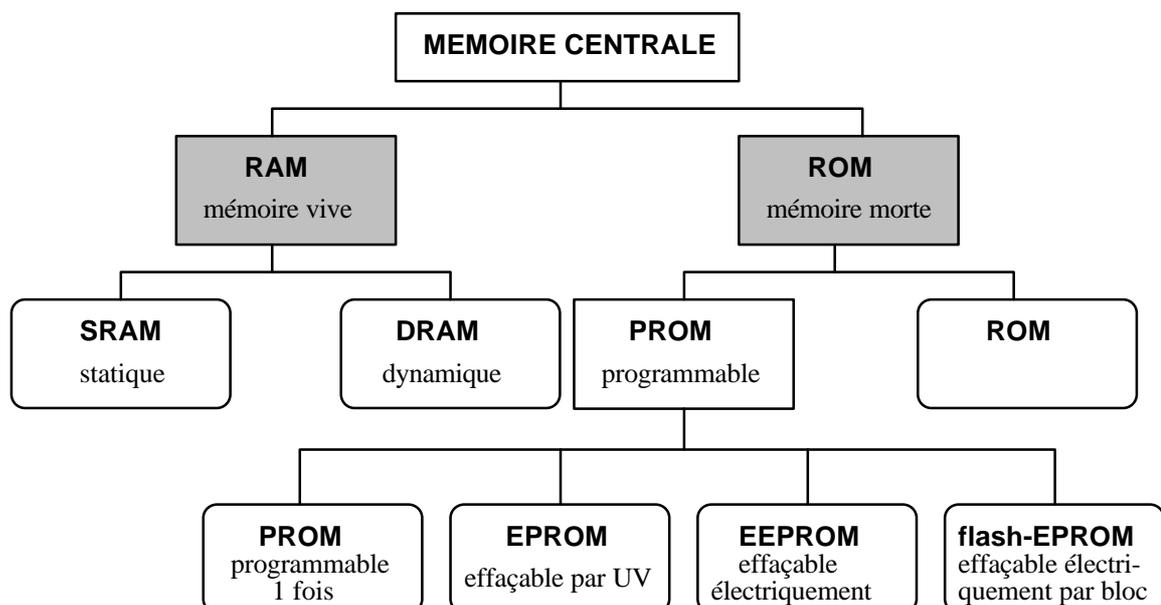
La mémoire centrale est la **zone de travail du processeur**, elle est inévitablement électronique puisque celui-ci est incapable de traiter directement les données sur support magnétique. La quantité de mémoire centrale utilisable est limitée par la **taille du bus d'adresses** (exprimée en nombre de bits) et les possibilités du **système d'exploitation**.

Tout au long de l'histoire de l'informatique, différentes **technologies** ont été utilisées pour la mémoire centrale (lignes à retard, mémoires à tores, mémoires à bulles) jusqu'aux **circuits intégrés** (héritiers des transistors des années 70).

CARACTERISTIQUES D'UN COMPOSANT DE MEMOIRE

- ☞ **Temps d'accès** (en nanosecondes) : temps entre le moment où le processeur accède à la mémoire et celui où l'information est disponible.
- ☞ **Fréquence d'accès** = inverse du temps d'accès. Par exemple, 20 MHz = $(1/50 \text{ ns}) \times 100$.
- ☞ **Temps de cycle** (= temps d'accès + temps de rafraîchissement) : temps minimum entre le début d'une lecture (ou écriture) et le début de l'opération suivante.
- ☞ **Capacité en nombre de bits**.
- ☞ **Matière** : **silicium** dans 90% des cas, mais certains ordinateurs emploient l'**arséniure de gallium** qui est 5 à 6 fois plus rapide (sur le Cray).
- ☞ **Volatilité** : perte de l'information si une coupure d'alimentation survient (RAM).

DIFFERENTS TYPES DE RAM ET ROM



TECHNOLOGIE : LES DIFFERENTS TYPES DE ROM

DEFINITION

La **ROM** (Read Only Memory), dite **mémoire morte**, est une mémoire non volatile (rémanente) qui n'autorise (en général) **que la lecture** de données qui y ont été introduites définitivement.

Une ROM contient informations nécessaires au bon fonctionnement interne d'un ordinateur ou d'un périphériques (routines systèmes, paramètres de configuration).

DIFFERENTS TYPES DE ROM

☞ **ROM "classique"** (1969)

C'est une mémoire dont le contenu est écrit directement par le fabricant du composant, d'après les indications du constructeur de l'ordinateur. Les données contenues en ROM ne sont accessibles qu'en lecture par l'utilisateur de l'ordinateur.

☞ **PROM** (Programmable ROM) ou **OTP-EPROM** (1969)

C'est une ROM non programmée par le fabricant et livrée vierge. Elle ne peut être écrite qu'une fois par le créateur du programme qu'elle va contenir (à l'aide d'un programmeur électrique de PROM).

☞ **EPROM** (Electrically Programmable ROM) ou **REPROM** (Reprogrammable ROM) ou **UV-EPROM** (1969)

C'est une mémoire réinscriptible plusieurs fois par programmation électrique, après que le contenu ait été effacé par exposition à la lumière ultraviolette. Elle est utilisée pour le test et la mise-au-point des programmes à porter sur des ROM.

☞ **EEPROM** (Electrically Erasable Programmable ROM) ou **EAROM** (Electrically Alterable ROM) (1980)

Le contenu de ce type de mémoire est modifiable par l'envoi d'un courant électrique, sans démontage de la carte mère ou de la carte d'extension. Cependant, on ne peut écrire les informations qu'en une seule fois et il faut toujours effacer les données avant d'en écrire de nouvelles, ce qui rend l'EEPROM plus lente que la RAM. Ces mémoires sont utilisées pour la configuration des **cartes d'extension** par logiciel.

☞ **flash EPROM** ou **mémoire Flash** (1988)

Cette mémoire est constituée de composants effaçables à partir du système d'un ordinateur, mais par blocs entiers. Une mémoire flash facilité les mises-à-jour et dispose d'une plus grande capacité que pour les EEPROM (pour un prix au bit moins élevé).

Cette technologie est utilisée dans les BIOS et les **cartes d'extension au format carte de crédit (PCMCIA)**.

TECHNOLOGIE : ROM D'UN COMPATIBLE PC

CONTENU DE LA ROM D'UN COMPATIBLE PC

☞ **ROM-BIOS** (Basic Input Output System) implantée de l'adresse FE000h à FFFFFh. La **ROM-BIOS** est un ensemble de micro-programmes (**firmware**) stockés dans une ROM, qui assurent les fonctions de base de l'ordinateur et dont l'accès est réalisé aux moyens **d'interruptions** (matérielles ou logicielles). Il ne faut pas confondre la **ROM-BIOS** (en ROM) et le **DOS-BIOS** qui fait partie de **MS/DOS** (en RAM et sur disque).

☞ **Table des caractères.**

CONTENU DE LA ROM-BIOS

☞ **Mini auto-test** (POST : Power-On Self Test) :

- Contrôle de la configuration de l'ordinateur (processeur, ROM-Bios, RAM, clavier, souris, disques, lecteur de disquettes).
- Initialisation des zones de communications des composants matériels et de la table des vecteurs d'interruptions, puis écriture dans celle-ci des informations correspondant aux vecteurs de la ROM-BIOS.
- Lancement des éventuels BIOS complémentaires (liés à des extensions) détectés sur le bus système (cartes graphiques, réseau, contrôleur SCSI,...).

☞ **Programme de chargement du BOOT** (programme d'amorçage) qui se trouve sur le secteur 1 de la piste 0 (1er secteur logique) du disque ou de la disquette.

☞ **Gestionnaire de services de bas niveau** : sur demande d'un programme, ou du système, le gestionnaire assure certains services d'entrée/sortie en rapport avec les périphériques (effacement d'écran, passage du mode texte au mode graphique, lecture disque, impression,...) ; ces fonctions de bas niveau adressent directement le matériel.

☞ **Informations diverses** : date du BIOS située en F000:FFF0, octets d'identification ("FF" pour un PC, "FE" pour un XT, "FC" pour un AT,...) situés en FFFFE.

REMARQUES

Le **BIOS** peut être **spécifique à un constructeur** (les grands constructeurs IBM, Compaq, HP, ZDS,... l'adaptent au mieux à leur machine) ou de type "**prêt à porter**" (acheté chez AMI, Phoenix, Award, Chips Technologies ou OPTI). Si les fonctionnalités sont les mêmes, les options, elles, diffèrent ; un éditeur peut ainsi proposer le même BIOS avec une option de sécurité par mot de passe, de nouveaux types de disques durs, des gestions de mémoires particulières ou même un antivirus intégré (BIOS Award).

afpa ©	auteur	centre		formation	module	séq/item	type doc	millésime	page 8
	M. ASTRE	Beaumont					sup. form.	05/97 - v2.0	MEMOIRE

TECHNOLOGIE : RAM

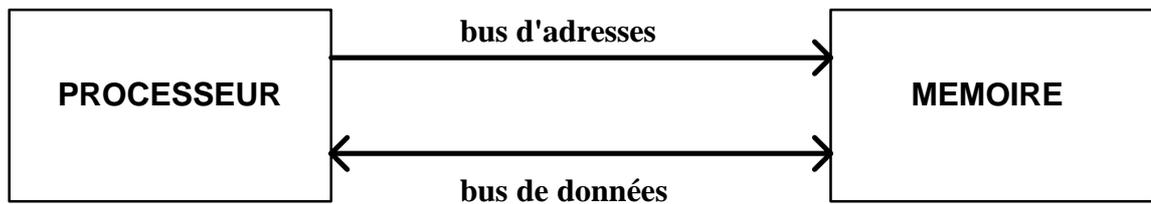
DEFINITION

La **RAM** (Random Access Memory), dite mémoire à accès aléatoire ou **mémoire vive**, est une mémoire volatile qui peut être lue et écrite.

La mémoire vive est donc **l'espace de travail du processeur**. Le processeur travaillant séquentiellement (une opération après l'autre, sur une donnée après l'autre), il doit s'entourer d'auxiliaires capables de **stocker** :

- ☞ les **instructions** du programme en cours d'exécution,
- ☞ les **données** manipulées par les instructions,
- ☞ les **résultats** de son travail,

BUS D'ADRESSES ET DE DONNEES



☞ Bus d'adresses

Le processeur envoie une requête de recherche d'un octet (pour un ordinateur 8 bits) par l'intermédiaire du bus d'adresse vers les 8 composants mémoire contenant les bits correspondants (repérage des bits à lire ou à écrire).

☞ Bus de données

Le transfert des informations entre processeur et composants de RAM est assuré par le bus de données.

TECHNOLOGIE : LES DIFFERENTS TYPES DE RAM

DRAM : RAM dynamique

☞ **Utilisation** : RAM traditionnelle (mémoire volatile).

☞ **Vitesse** : de 50 à 70 ns.

☞ **Composition** : Les cellules mémoires sont organisées en lignes et en colonnes à l'intersection desquelles se trouvent des transistors et des condensateurs. La présence ou non d'une charge électrique dans un **condensateur** est interprétée comme un 1 ou un 0. Les transistors font eux office de vannes chargées de réguler le trafic.

En fait, l'adressage s'effectue par **case mémoire de 8, 16 ou 32 bits**. C'est-à-dire qu'à chaque case mémoire correspond un paquet de condensateurs fonctionnant de concert.

☞ **Conservation des données** : quelques millisecondes sans rafraîchissement. Les charges électriques ont tendance à diminuer dans le temps jusqu'à disparaître, ce qui oblige l'ordinateur à les "**rafraîchir**" périodiquement (tous les 1 à 2 millièmes de seconde), c'est à dire à recharger les condensateurs par de nouvelles impulsions électriques, pour éviter les pertes d'information. Ce mouvement électrique permanent justifie le terme de "dynamique".

☞ **Avantage** : les DRAM offrent une capacité 4 fois supérieure aux produits statiques (à degré d'intégration égal) et sont donc bien plus économiques.

☞ **Inconvénients** : volatilité et existence de temps d'attente lié au rafraîchissement.

DIFFERENTES TECHNOLOGIES DE DRAM

La technologie de RAM détermine le type d'adressage des cellules mémoires. Pour utiliser une technologie de RAM particulière, il faut que la carte mère l'accepte.

Type de RAM	FPM	EDO	SDRAM	EDRAM	MDRAM	RDRAM
temps d'accès	60 ns	60 ns	60 ns	30 ns	30 ns	60 ns
fréquence max.	25 MHz	50 MHz	100 MHz	83 MHz	167 MHz	300 MHz
taux transfert	40 Mo/S	66 Mo/s	132 Mo/s	83 Mo/s	660 Mo/s	600 Mo/s
utilisation	ancien standard	standard depuis 1996	futur standard ?			

☞ **FPM** (Fast Page Mode)

En mode "classique", le processeur adresse chaque bit dont il a besoin (une instruction ou une donnée peuvent représenter des centaines de bits situés séquentiellement dans la RAM). En mode page, on déclare une adresse de début et une adresse de fin entre lesquelles on lit tous les bits. Lorsqu'un accès a lieu dans une page donnée, tous les accès suivants dans la même page se font sans état d'attente.

☞ **EDO** (External data Out)

Une mémoire EDO complète la DRAM avec une petite mémoire tampon permettant de charger l'adresse de la prochaine donnée à lire ou à écrire sans attendre que la donnée précédente soit lue.

☞ **SDRAM** (Synchronous DRAM)

Une mémoire SDRAM possède sa propre horloge interne au lieu d'utiliser le signal en provenance du processeur. Les opérations de lecture/écriture sont rendues indépendantes afin de permettre le traitement simultané de plusieurs accès. Elle offre un gain de performances pouvant aller jusqu'à 20% par rapport à l'EDO.

☞ **EDRAM** (Enhanced DRAM) de Ramtron et **CDRAM** (Cached DRAM) de Mitsubishi : RAM comportant une petite quantité de cache SRAM (quelques Ko) dans ses circuits.

☞ **MDRAM** (Multibank DRAM) de Mosys : plusieurs banques indépendantes permettent d'accéder à plusieurs adresses simultanément pour peu que celle-ci concernent différentes banques.

☞ **RDRAM** (Rambus DRAM) : DRAM augmentant la fréquence jusqu'à 300 MHz.

SRAM : RAM statique

☞ **Utilisation** : dispositifs de mémoire **cache** (RAM rapide entre processeur et RAM traditionnelle).

☞ **Vitesse** : de 8 à 20 ns.

☞ **Conservation des données** : Contrairement aux DRAM, les modules de SRAM n'ont besoin ni d'être réécrits, ni d'être rafraîchis et peuvent conserver indéfiniment l'information (tant qu'ils sont alimentés).

☞ **Avantage** : pas de temps d'attente (zéro wait state), le temps de cycle est égal au temps d'accès.

☞ **Inconvénients** (par rapport aux DRAM) : consommation électrique supérieure et intégration inférieure (là où un composant de DRAM stocke 1 million de bits, une SRAM en contiendra difficilement plus de 256000 sur la même surface de silicium). La SRAM est donc plus chère que la DRAM.

RAM VIDEO

☞ **Architecture « classique »** : les **cartes vidéo** comportent de la DRAM chargée de stocker le contenu de l'écran ; c'est ce qu'on appelle la **mémoire vidéo**. Les cartes vidéo de haut de gamme peuvent utiliser des variantes de DRAM.

- **VRAM** (RAM Vidéo) : RAM statique de type "dual-ported", c'est-à-dire que l'on peut y lire et y écrire en même temps (contrairement à la DRAM et à l'EDO qui sont de type "single-ported").
- **WRAM** (Windows RAM) : mémoire vidéo "dual-ported" accélérant les déplacements de fenêtres.
- **SGRAM** (Synchronous Graphic RAM) : principe des SDRAM appliqué aux cartes graphiques.

☞ **Architecture UMA** (Unified Memory Architecture) utilisée par certains micros.

Dans l'architecture UMA, le contrôleur graphique utilise une portion de la RAM standard comme mémoire vidéo. Cette technique permet de faire des économies sur les cartes graphiques, mais elle est réservée aux machines dotées d'une mémoire de grande capacité. UMA existe sous forme de spécification **SMBA** (Share Memory Buffer Architecture) d'Intel et de la norme **VUMA** du comité Vesa.

TECHNOLOGIE : RAM SETUP D'UN COMPATIBLE PC

SETUP

Sur les premiers PC, les **paramètres de configuration** (taille RAM, présence du coprocesseur, nombre et type de mémoires auxiliaires) étaient déterminés par l'intermédiaire d'une série de **switches** situés sur la carte mère (ce qui limitait le nombre de paramètres).

A partir de l'AT, les ordinateurs furent dotés d'une petite **RAM CMOS** (64 octets) entretenue sur une **pile** autonome (d'une durée de vie de 2 à 8 ans) destinée d'abord à conserver la date et l'heure, puis les paramètres de configuration. L'accès à cette mémoire se fait à l'aide d'un utilitaire de configuration (**Setup**) souvent **incorporé à la ROM-Bios** des ordinateurs. Cet utilitaire **se lance par une combinaison de touches** utilisables à tout moment, ou seulement au **boot** (telles que Del, F1, Esc, Ctrl+Esc, Alt+Esc, Ctrl+Alt+Enter, Shift+Ctrl+Alt+Del,...). Dans de rares cas, l'utilitaire de configuration est livré sur disquette, il se lance alors par une commande telle que "Setup". Le menu du Setup peut aussi **s'afficher automatiquement**, à la mise sous tension, s'il y a un **problème** de configuration.

PARAMETRES DU SETUP

Les options proposées par un Setup, ainsi que leur présentation, dépendent du Bios.

- ☞ Date et heure.
- ☞ Mémoire de base et étendue.
- ☞ Mémoire de masse : type de lecteur de disquette et de disque dur (nombres de têtes, de cylindres, de secteurs/piste), ordre de sollicitation des drives au démarrage (A: puis C:, C: puis A:).
- ☞ Mémoire cache du processeur, vitesse d'horloge du bus.
- ☞ Type de contrôleur écran utilisé.
- ☞ Mot de passe (saisi en QWERTY, car au moment du contrôle la table de gestion du clavier français ne sera pas encore chargée). En cas d'oubli du mot de passe, débrancher la batterie qui alimente la mémoire CMOS et attendre un certain temps afin que les informations correspondantes se volatilisent, puis rebrancher la batterie, accéder au Setup et resaisir la configuration.
- ☞ Anti-virus.
- ☞ Mémoire shadow (recopie du BIOS en RAM).

REMARQUES

Le Setup s'utilise en cas de **changement de configuration**, tels qu'ajout de RAM ou remplacement de disque (certains Bios détectant automatiquement ces changements).

Afin de **garder une trace de la configuration**, il est conseillé de faire une copie de l'écran du setup avant de modifier les paramètres (ou pour prévenir une panne de la pile).

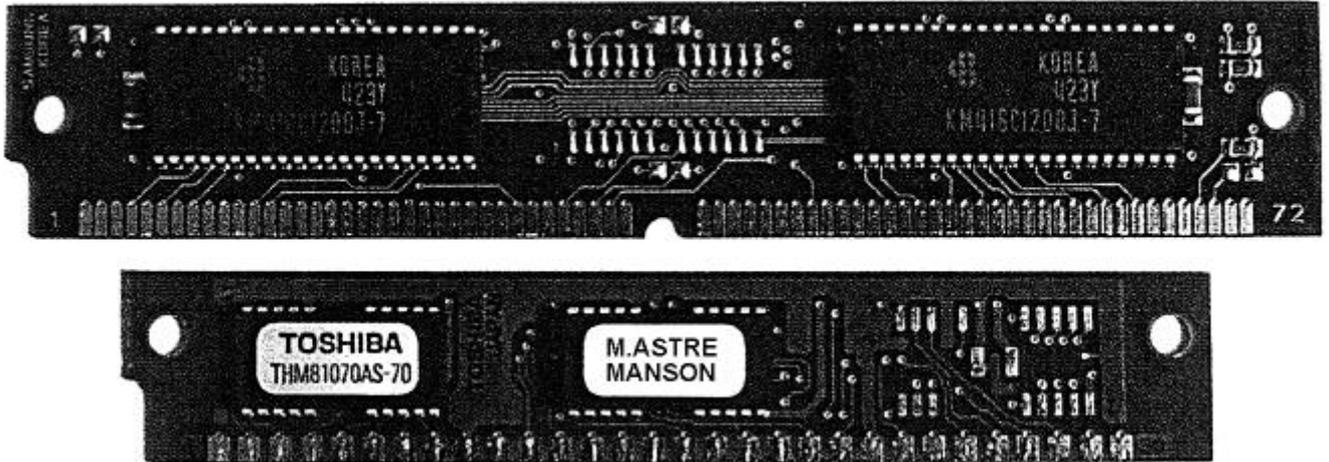
afpa ©	auteur	centre		formation	module	séq/item	type doc	millésime	page 12
	M. ASTRE	Beaumont					sup. form.	05/97 - v2.0	MEMOIRE

TECHNOLOGIE : INSTALLATION DE RAM SUR UN PC

SUPPORTS DE RAM

☞ **Barettes SIMM** (Single In-line Memory Module) : occupent la quasi-totalité du marché depuis 1991.

Une barrette SIMM comporte plusieurs **puces de RAM** soudées et peut être utilisée pour la **mémoire centrale**, mais aussi pour la **mémoire vidéo** et certaines **imprimantes**.



Les barrettes de RAM sont regroupées au sein de **banques** à l'intérieur desquelles elles doivent avoir théoriquement les **mêmes caractéristiques**, bien qu'en pratique on puisse se permettre quelques transgressions. De plus, les modules SIMM sont **standards** mais le nombre de barrettes, ainsi que leurs caractéristiques, varient en fonction des micros.

- **Type de mémoire** : **9 bits** (286 et 386SX) ou **36 bits** (486 et Pentium). Quand il n'y a pas bit de parité (notamment pour les Macintosh), on parle alors de module de **8, 32 ou 64 bits** (Pentium Pro).
- **Technologie** : FTP, EDO, SDRAM.
- **Capacité d'une barrette** : 1, 2, 4, 8, 16, 32 ou 64 Mo.
- **Temps d'accès** : de 60 à 100 ns (la moyenne se situant à 70 ns).

Le **prix moyen d'une barrette** est de 500 F pour une barrette de 16 Mo en 32 bits, et de 1 500 F pour une barrette de 32 Mo (janvier 97). Cependant, il existe une grande différence de prix (de l'ordre de 20%) entre les barrettes signées d'un grand constructeur (IBM, HP, Compaq,...) et son équivalent vendu sous la marque d'un fabricant de mémoires compatibles.

☞ **Modules DIMM** (Dual In-line Memory Module) : apparus en 1996 et devraient remplacer les SIMM 32 bits. Cependant, ils souffrent d'un manque de standardisation.

Caractéristiques : 64 bits, 128 broches, adaptés au bus PCI.

☞ **Cartes d'extension mémoire au format carte de crédit (PCMCIA).**

EXTENSION DE LA MEMOIRE

☞ Installation de barrettes SIMM :

Une barrette SIMM possède un connecteur de type bord de carte et s'enfiche soit sur la carte mère, soit sur une carte d'extension "propriétaire" acceptant ce type de barrettes.

☞ **Configuration du système** (avant de remonter le boîtier afin de vérifier le bon fonctionnement) : une fois l'extension mémoire installée, il faut en avertir l'ordinateur en lançant le Setup au démarrage de l'ordinateur (s'il ne se lance pas automatiquement). Il faut toutefois remarquer que certaines cartes mère comprennent des cavaliers pour déclarer la taille mémoire.

afpa ©	auteur	centre		formation	module	séq/item	type doc	millésime	page 14
	M. ASTRE	Beaumont					sup. form.	05/97 - v2.0	MEMOIRE

MEMOIRE DU PC : LIMITES DES COMPATIBLES PC

GESTION DE LA MEMOIRE PAR LES MICROPROCESSEURS

processeurs	8086	80286	80386SX	80386	486	Pentium
bus adresses	20 bits	24 bits	24 bits	32 bits	32 bits	32 bits
bus données (bus mémoire)	16 bits	16 bits	16 bits	32 bits	32 bits	64 bits
capacité de gestion mémoire directe	1 Mo	16 Mo	16 Mo	4 Go	4 Go	4 Go
capacité de gestion mémoire virtuelle	1 Mo	1 Go	64 To	64 To	64 To	64 To

L'espace adressable est fonction de la largeur du bus d'adresse (20 lignes donnent 1 Mo sur un 8086, 32 lignes donnent $2^{32} = 4$ Go sur un 80386).

GESTION DE LA MEMOIRE PAR MS/DOS

☞ **MS/DOS est limité à 640 Ko.**

Le **MS/DOS, conçu au départ pour les processeurs 8086/8088** (pouvant adresser **1 Mo**), **a hérité de leur limite**. Cette limite ne semblait pas pénalisante lors de la sortie du premier IBM PC en 1981. En effet, le standard de l'époque, le système d'exploitation CP/M, était limité à un espace de 64 Ko, et les premiers PC ne disposait que de 64 Ko de RAM. Le MS/DOS ne peut donc **gérer directement que 640 Ko de RAM** utilisateur (le complément jusqu'à 1 Mo étant réservé).

☞ **Aujourd'hui, cette limite est pénalisante.**

- La mémoire est de plus en plus encombrée de **résidents** et de **drivers** (réseau, cache, anti-virus,...).
- L'utilisateur travaille sur des **documents** de plus en plus **volumineux** (documents composites avec texte, image,...).
- **Les logiciels sont de plus gourmands en mémoire.** En 1983, un traitement de texte fonctionnait avec seulement 64 Ko. Aujourd'hui, Word réclame 8 Mo.

☞ **Au delà des 640 Ko du DOS, la mémoire supplémentaire est gérée en mémoire étendue XMS.**

☞ **Tailles mémoire nécessaires :**

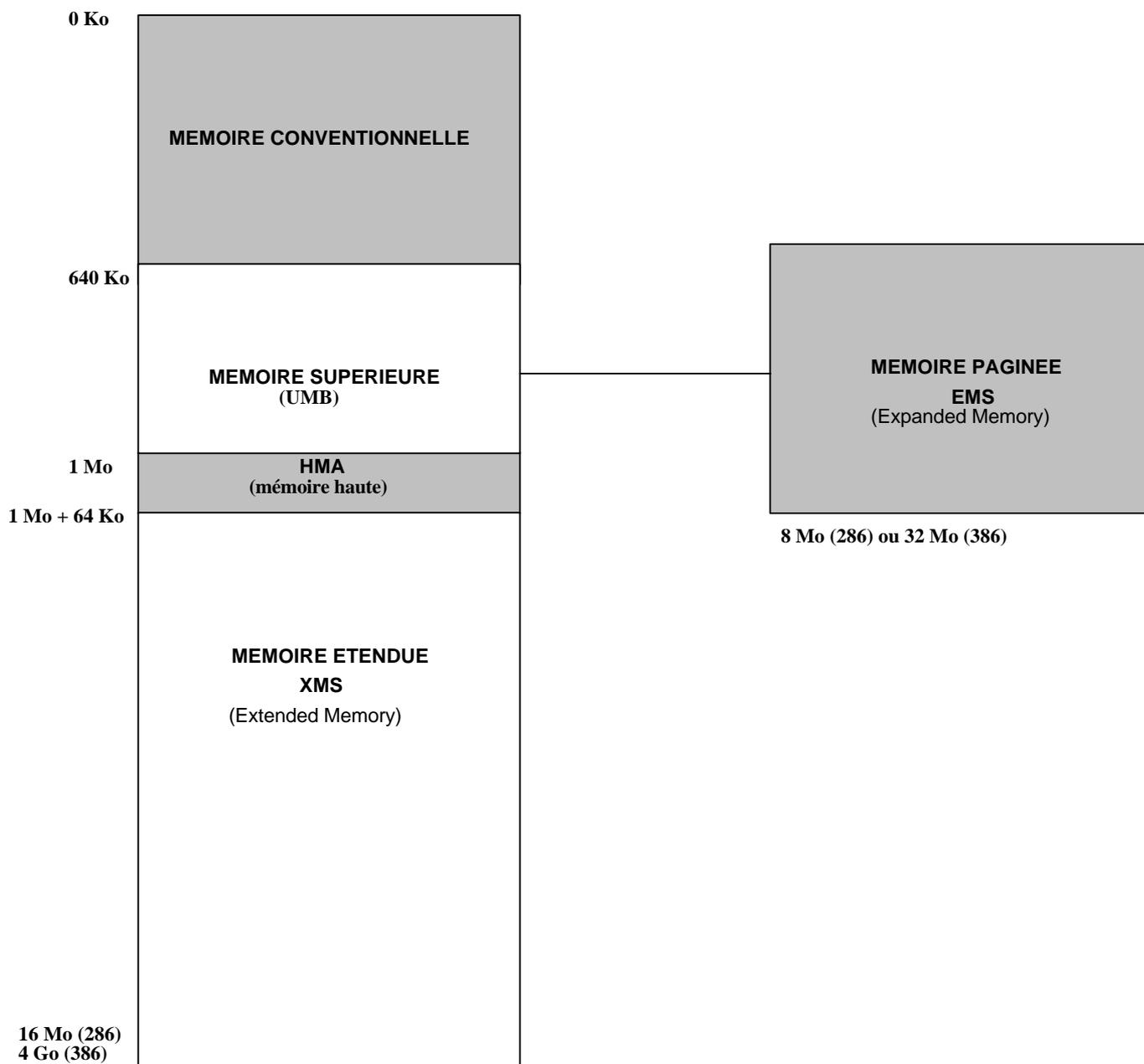
PC en mode texte	: 1 Mo
Windows 3.1 ou Système 7	: 8 Mo (4 Mo minimum)
Windows 95	: 16 Mo (8 Mo minimum)
Windows NT	: 32 Mo (16 Mo minimum)

QU'EST-CE QUI LIMITE L'UTILISATION DE LA MEMOIRE ?

☞ **Le matériel :** capacité d'adressage du **processeur, BIOS** et possibilités d'extension de la **carte mère**.

☞ **Le logiciel :** **système d'exploitation** (640 Ko pour MS/DOS) et **interface graphique** (16 Mo pour Windows 3.1)

MEMOIRE DU PC : STRUCTURE GENERALE DE LA MEMOIRE



La différence entre les divers types de mémoire est purement logique. Physiquement, les barrettes de RAM utilisées sont les mêmes.

type de mémoire	conventionnelle	supérieure	haute (HMA)	étendue XMS	paginée EMS
type de micro	tous	386 (MS/DOS) 286 (utilitaire)	286 et plus	286 et plus	tous
maximum	640 Ko	180 Ko	64 Ko	16 Mo (286) 4 Go (386)	8 Mo (286) 32 Mo (386)
gestionnaire		EMM386 + HIMEM.SYS	HIMEM.SYS	HIMEM.SYS	EMM386 ou driver EMM
mode processeur	réel	réel	réel	protégé	réel par simulation

MEMOIRE DU PC : MEMOIRE CONVENTIONNELLE (de 0 à 640 Ko)

OCCUPATION DE LA MEMOIRE CONVENTIONNELLE (de 0 à 640 Ko) SOUS MS/DOS

C'est dans cet espace mémoire que s'exécutent les programmes utilisateurs et les commandes externes de MS/DOS.

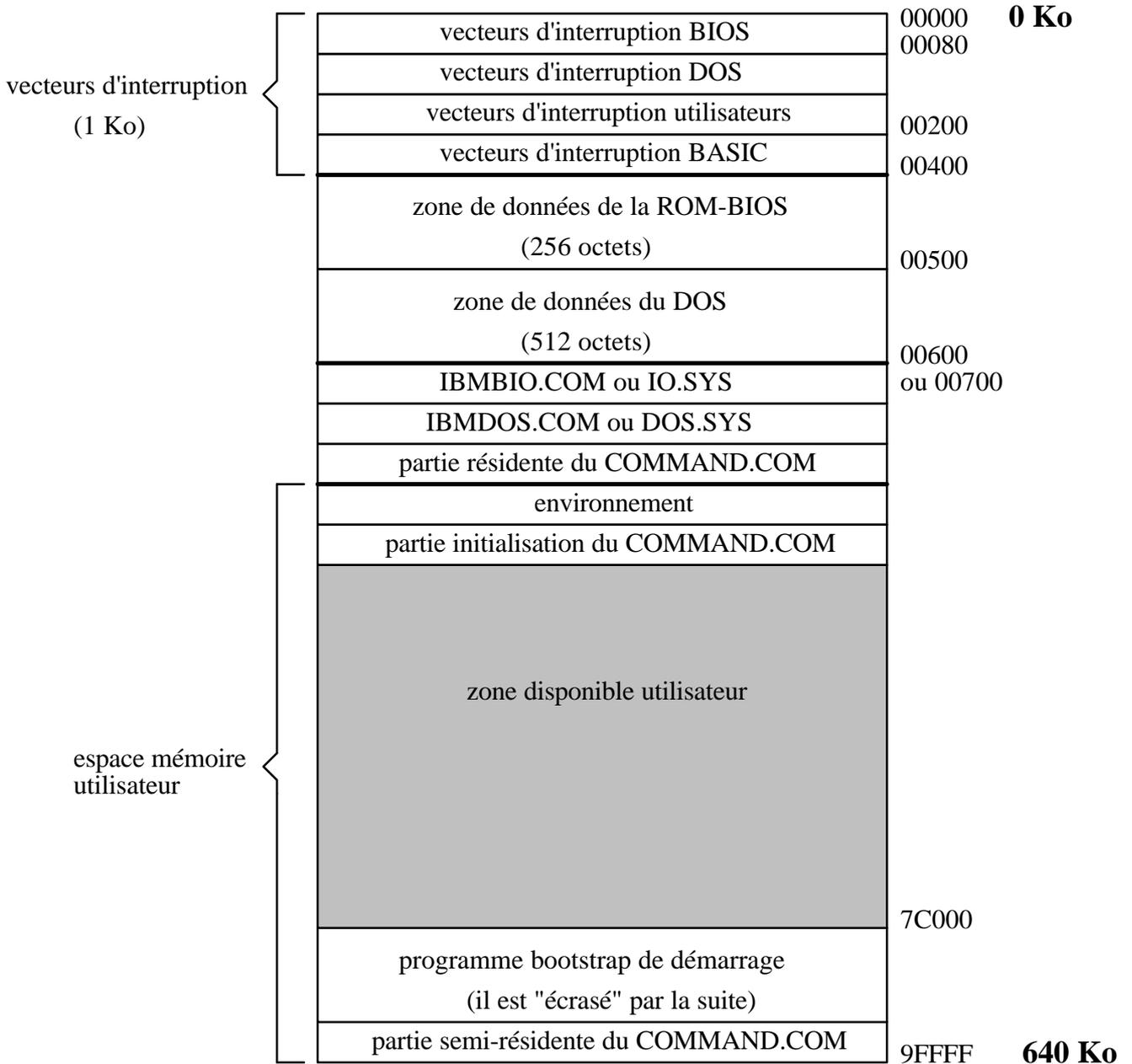


TABLE DES VECTEURS D'INTERRUPTIONS

☞ Cette table permet de localiser le **gestionnaire** associé à chaque **interruption**.

Une **interruption** est une **requête** adressée au **processeur** de la part d'un composant **matériel** (clavier, souris, port série,...) ou d'un **programme**.

Lorsqu'une interruption arrive au processeur, celui-ci suspend provisoirement son traitement afin de s'occuper de cette requête. En fonction du signal activé, il déclenche un **programme (gestionnaire d'interruption)** pour traiter cette interruption. Lorsque cette **routine** est terminée, le programme interrompu reprend.

☞ Le terme "**vecteur**" est utilisé pour désigner **l'adresse** (sur 4 octets) du programme de gestion d'interruption.

Le vecteur d'interruption n° **n** se trouve à la position mémoire "**4 X n**".

E8	4E	9A	01	se lit 019A:4EE8
----	----	----	----	------------------

L'adresse est stockée dans **l'ordre inverse** :

☞ Il y a **3 types d'adresse** :

- Sur la **ROM-BIOS** ('F' en début de segment).
- Sur la **mémoire RAM** (sous-programme du DOS ou programme résident).
- **0**, si l'interruption n'est pas gérée.

ENVIRONNEMENT : zone de mémoire qui contient les **variables d'environnement** (PATH, PROMPT, COMSPEC, variables définies avec SET).

CONSTITUTION DU COMMAND.COM (interpréteur de commande de MS/DOS)

☞ **Partie résidente** située après le DOS : modules de traitement des interruptions 22h, 23h et 24h, programme de rechargement de la partie semi-résidente, gestions d'erreurs et messages d'erreur, commandes internes, gestion CTRL+C.

☞ **Partie initialisation** qui ne sert qu'à la phase de démarrage du micro-ordinateur, la place est donc libérée après son exécution (principe de recouvrement dit « overlay ») : programme de chargement d'AUTOEXEC.BAT, affichage de la date, détermination de l'adresse d'implantation où le système d'exploitation chargera les programmes avant de les exécuter.

☞ **Partie semi-résidente** concernant la gestion de fichiers de commande « .BAT ». Cette partie peut donc être écrasée si le système d'exploitation a besoin de place, elle sera rechargée lorsque le processeur de commande reprendra la main (partie résidente de COMMAND.COM).

RESSOURCES SYSTEME

Certains éléments de configuration prennent de la place en mémoire :

- ☞ Liste des disques : **LASTDRIVE** (80 octets/lecteur au-delà de E)
- ☞ Taille des piles : **STACKS**
- ☞ Nombre de fichiers ouverts déclarés : **FILE** (64 octets/file)
- ☞ Taille des tampons : **BUFFER** (532 octets/buffer)
- ☞ Cache de répertoire : **FASTOPEN** (48 octets/fichier)
- ☞ File Control Block : **FCBS** (64 octets/fichier)

MEMOIRE DU PC : RESIDENTS (TSR) ET DRIVERS

PROGRAMME RESIDENT (TSR : Terminate and Stay Resident)

☞ Un résident est un logiciel "dormant" (ou fonctionnant en « tâche de fond ») en mémoire centrale et **activable** (par interruptions) à **volonté** par une combinaison de touches (SIDEKICK,...) **ou dans des situations données** (KEYB,...).

Normalement, lorsqu'un programme se termine, le système ferme les fichiers que ce programme aurait oublié de fermer, puis nettoie la mémoire afin de la rendre pour la prochaine application. Mais un résident termine en disant au DOS "je reste en mémoire, et je continue d'occuper telle place" : le programme se termine et reste résident (**TSR = Terminate and Stay Resident**).

☞ **Programmes résidents sous MS/DOS :**

- **Résidents systèmes indispensables :**

- Programmes système **IO.SYS** et **DOS.SYS**.
- Interpréteur de commandes **COMMAND.COM**.
- Driver de clavier français **KEYB.COM**.

- **Résidents systèmes facultatifs**, par exemple **SHARE** permet le partage des fichiers.

- **Bibliothèques dynamiques** : programmes qui restent en mémoire et fournissent l'ensemble des fonctions et des sous-programmes nécessaires à l'exécution de certaines applications compilées. Par exemple **BASRUN** pour certains compilateurs BASIC.

- **Logiciels de télécopie ou de communication.**

- **Utilitaires.** Par exemples, **NORTON COMMANDER** ou **PC-TOOLS**.

☞ **Remarques**

- Evaluation de la taille d'un résident : **MEM/C**.

- Les résidents s'empilent en RAM. Si l'on doit **désactiver un résident**, il est indispensable de mettre au préalable hors fonction ceux qui ont été chargés après lui (et tous les résidents ne sont pas désactivables). Par exemple, les résidents réseau Netware peuvent se décharger avec le paramètre « /u » : **LSL/U**, **IPXODI/U**, etc.

- La fonction d'accès temporaire au DOS de certains logiciels (Word, Paradox,...) peut être **incompatible avec certains résidents**. De plus certains résidents ne fonctionnent pas sous le mode protégé de Windows.

- Les résidents peuvent être **relogés en mémoire supérieure** (voir "Utilitaires de gestion mémoire").

GESTIONNAIRE DE PERIPHERIQUE (dit « DRIVER » ou « PILOTE »)

Un driver est un petit programme qui pilote un périphérique particulier.

IO.SYS contient des drivers pour le clavier, le système d'affichage, les ports série et parallèle. Ces drivers chargés par défaut sont les suivants :

driver	description
A: à I:	unités de disque
AUX	périphérique auxiliaire
CLOCK	horloge système
COM	ports série (COM 1 à 4)
CON	clavier/écran
LPT	ports parallèles (LPT 1 à 3)
NUL	périphérique nul
PRN	imprimante (en général LPT1)

Bien que le DOS fournisse des drivers, il autorise l'ajout de nouveaux drivers et le remplacement des siens. Exemples de **drivers fournis avec le DOS** :

driver	description
ANSI.SYS	remplace CON par un driver de clavier/écran plus sophistiqué
DISPLAY.SYS	permet la commutation de pages de codes pour l'affichage et le clavier
EMM386.EXE	sert à créer de la mémoire paginée à partir de mémoire étendue, et à accéder aux UMB
HIMEM.SYS	gestionnaire de mémoire étendue
RAMDRIVE.SYS	créer un disque virtuel
SETVER.EXE	sert à changer artificiellement de version de DOS afin de pouvoir exécuter certaines applications
SMARTDRV.SYS	cache

Exemples de **drivers fournis avec un périphérique** : souris (MOUSE.SYS), streamer, disque SCSI, scanner, réseau local, lecteur CD,...

Du point de vue de l'utilisateur, la seule **différence entre un TSR et un driver** est le moment et la façon dont ils sont chargés en mémoire. Par exemple, pour la souris, on peut déclarer le **driver MOUSE.SYS** dans le **Config.sys** ou le **TSR MOUSE.COM** dans l'**Autoexec.bat**. Un **driver** est en général chargé à l'initialisation et ne peut pas être déchargé. Par contre, un **TSR** est chargé pendant ou après l'initialisation et il peut éventuellement être déchargé.



MEMOIRE DU PC : MEMOIRE SUPERIEURE (de 640 Ko à 1 Mo)

ESPACE MEMOIRE DE 1 Mo DU PC

☞ **Mémoire conventionnelle** (mémoire utilisateur) : 10 segments de 64 Ko (de 0 à 9) composent les **640 Ko** directement accessibles par MS/DOS.

☞ **Mémoire supérieure (de 640 Ko à 1 Mo)** : les 6 segments suivants (de A à F) sont réservés par le DOS pour des tâches particulières (système et ROM).

localisation physique	taille	ZONES MEMOIRE	adresse		n° de segment	
			en hexa	en Ko		
RAM sur la carte mère	640 Ko (10 x 64 Ko)	MEMOIRE CONVENTIONNELLE MS/DOS	00000	0 Ko	segment 0	
RAM sur carte vidéo	128 Ko	MEMOIRE SUPERIEURE		A0000	640 Ko	segment A
			mémoire vidéo	B0000	704	segment B
				C0000	768	segment C
				D0000	832	segment D
				E0000	896	segment E
				F0000	960	segment F
la localisation de la RAM dépend de l'extension	192 Ko	espace disponible pour des extensions	F3FFF			
	16 Ko	réservé	F4000	976		
ROM sur la carte mère	8 Ko	ROM				
		ROM de base (non utilisée)	F6000	984		
	32 Ko	ROM-Basic (interpréteur Basic)	FE000	1016		
	8 Ko	ROM-Bios	FFFFF	1 Mo		

MEMOIRE VIDEO

☞ **Mémoire vidéo de la carte monochrome (4 Ko).** La carte monochrome permet l'affichage uniquement en **mode texte** sur 25 lignes X 80 colonnes (2000 caractères). Chaque caractère affichable est composé de **2 octets** consécutifs. Le premier contient le code ASCII du caractère, le deuxième contient l'attribut de visualisation.

☞ **Mémoire vidéo de la carte couleur (16 Ko).** La carte couleur permet de travailler en 2 modes :

mode texte : 25 lignes X 80 colonnes

mode graphique : - soit 320 X 200 points en 4 couleurs (2 bits/point)
 - soit 640 X 200 points en 2 couleurs

☞ **Capacité de mémorisation de la mémoire vidéo :**

- en **mode texte** : en 25 X 80 : 4 pages écran (25 X 80 X 4 = 8000 octets)
 en 25 X 40 : 8 pages écran
- en **mode graphique** : en 320 X 200 : 1 page (320 X 200 X 2 bits = 128000 bits = 16000 octets)

☞ **Adresses de la mémoire vidéo** : par exemple ...

A000-AFFF	mémoire graphique
B000-B7FF	monochrome
B800-BFFF	mémoire graphique non monochrome
C000-C800	Rom-BIOS de la carte VGA

ESPACE DISPONIBLE POUR DES EXTENSIONS (C0000-F0000). Cet espace est occupé différemment suivant :

☞ Les **machines** : PC, XT, AT, 386/486 ou PS/2. Par exemple, sur les PS/2, le segment **E000** est occupé par des extensions spéciales du BIOS (mais ce segment est en général libre sur les autres micros). De plus, certaines versions du BIOS permettent l'utilisation de la première moitié du segment F (jusqu'à F7FF).

☞ Les **extensions** apportées :

- BIOS du contrôleur **disque dur** (C8000-C9FFF) pour un compatible XT (intégré dans le BIOS pour un AT).
- La mémoire **vidéo**.
- La zone d'échange **EMS** (64 Ko, de 784 Ko à 848 Ko).

Le reste est libre pour la **RAM utilisateur** (C0000, CC000, segments D0000 et E0000).

MEMOIRE DU PC : MEMOIRE PAGINEE EMS

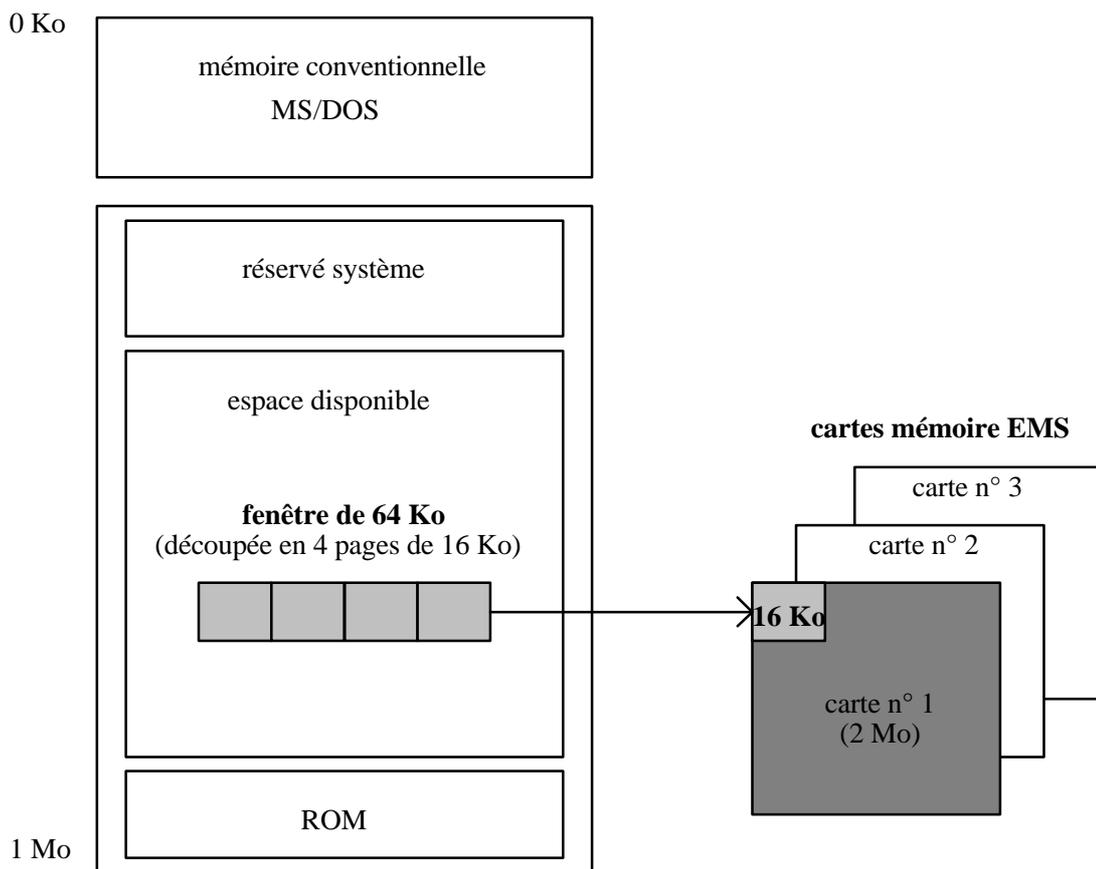
MEMOIRE PAGINEE (Expanded Memory : mémoire expansée) : obsolète

Attention, il ne faut **pas confondre** la **mémoire paginée EMS** avec la **mémoire virtuelle paginée** de Windows 95.

En 1985 certains constructeurs se sont réunis dans le cadre du **LIM** (Lotus-Intel-Microsoft) pour proposer une norme de mémoire étendue paginée référencée **EMS** (Expanded Memory Spécification).

FONCTIONNEMENT DE LA MEMOIRE EMS

Contrairement à la mémoire étendue en linéaire (qui utilise l'adressage direct), l'accès aux cartes mémoires additionnelles se fait par un **mécanisme de pagination mémoire logique vers une carte physique**. La mémoire EMS passe par une "fenêtre" (dite cadre de page, ou page frame) de 64 Ko (découpé en **4 pages de 16 Ko**), située en mémoire supérieure, qui joue le rôle de buffer pour les échanges entre la mémoire centrale et la mémoire EMS située sur une carte. Le processeur n'utilise donc qu'un ensemble d'adresses couvrant **64 Ko** pour adresser en fait une mémoire de **2 Mo**.



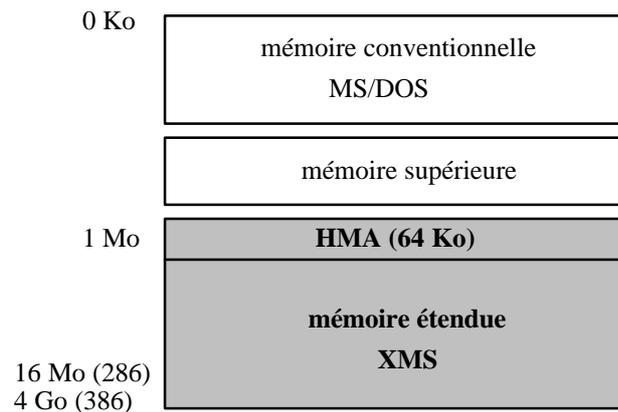
Ce type de mémoire est accessible en **mode réel** (contrairement à la mémoire étendue en linéaire qui nécessite le mode protégé). C'est donc la seule solution pour étendre la mémoire sur **8086** (adressage limité à 1 Mo), mais cette solution est aussi valable pour les compatibles AT.

MEMOIRE DU PC : MEMOIRE ETENDUE XMS

MEMOIRE ETENDUE (Extended Memory)

La mémoire étendue est une **extension linéaire de la mémoire au-delà de 1 Mo**. Cette mémoire est adressée directement par les mécanismes de segmentation. Le terme **EMB** (Extended Memory Block) désigne un bloc de mémoire étendue.

Ce type de mémoire n'est accessible qu'aux **programmes conçus pour l'utiliser**, tournant sur ordinateurs dotés d'un **processeur à partir du 80286** (mode protégé) et ayant un **gestionnaire de mémoire étendu**.



HMA : HIGH MEMORY AREA (à partir du 80286)

Un **8086** dispose de 20 lignes d'adresses ($2^{20} = 1 \text{ Mo}$) permettant un adressage jusqu'à FFFF:000F (=FFFF). Les adresses entre FFFF:0010 et FFFF:FFFF ne sont donc pas utilisées.

A partir du **80286** le nombre de lignes d'adresses augmente (24 pour le 286, 32 pour le 386). En activant la 21^{ème} ligne d'adresse (A20) les programmes qui s'exécutent en **mode réel** peuvent donc adresser la mémoire comprise entre FFFF:0010 et FFFF:FFFF (sans basculer en mode protégé).

Cette zone, appelée **HMA**, occupe les 64 premiers Ko (moins 16 octets) au dessus du premier Mo de mémoire. Le driver **HIMEM.SYS** (norme **XMS**) de Microsoft permet d'allouer et de désallouer de la mémoire dans cette zone HMA. De plus, à partir du DOS 5, il est possible de charger le **DOS** en HMA.

NORME XMS (eXtended Memory Spécification) de Lotus-Intel-Microsoft-AST (norme introduite en 1988 avec Windows 2.1)

Cette norme est destinée à gérer les accès à la **mémoire étendue linéaire** (en mode protégé) et à la **mémoire HMA** (comme zone d'échange entre la mémoire conventionnelle et la mémoire étendue). Le driver **HIMEM.SYS** (de MS/DOS et Windows) répond à cette norme XMS.

UTILISATION DE LA MEMOIRE ETENDUE

Un programme sous MS-DOS ne peut pas s'exécuter en mémoire étendue, ni y stocker des données, son utilisation est donc limitée : caches disques, disque virtuel, buffer ou spooler d'imprimante. La mémoire étendue linéaire n'est reconnue que par des **systèmes d'exploitation évolués** (Unix, OS/2, Windows NT) et par **certains logiciels** (Windows, Oracle, Informix,...).

afpa ©	auteur	centre		formation	module	séq/item	type doc	millésime	page 24
	M. ASTRE	Beaumont					sup. form.	05/97 - v2.0	MEMOIRE

OPTIMISATION MEMOIRE : UTILITAIRES DE GESTION MEMOIRE

FONCTIONS OFFERTES PAR UN GESTIONNAIRE DE MEMOIRE

fonctions	utilitaires
chargement en mémoire supérieure (utilisation des UMB) : - drivers - résidents	utilitaires DOS : EMM386 + ... DEVICEHIGH (Config.sys) LH (Autoexec.bat) autres utilitaires (QEMM386, 386Max)
compatibilité avec les spécifications XMS (mémoire étendue)	HIMEM.SYS, QEMM386, 386Max
recopie de ROM en RAM (mémoire fantôme)	BIOS, utilitaires (QEMM386)
cartographie de la mémoire	MEM, MSD, MFT
optimisation mémoire	MEMMAKER (auto), MFT (manuelle)

CHARGEMENT DE LOGICIELS EN MEMOIRE SUPERIEURE

☞ La **mémoire supérieure** (espace mémoire entre 640 Ko et 1 Mo) est réservée à des tâches particulières et son occupation est fonction de la configuration de chacun. Toutefois il subsiste des "**étages vides**" dans l'espace d'adressage (**UMB** : Upper Memory Blocks). Mais ces "trous" virtuels ne peuvent à eux seuls contenir des informations, ils doivent être comblés avec de la vraie mémoire, et ne peuvent être manipulés directement, il faut faire appel à des utilitaires afin d'y charger les résidents.

☞ Donc, **pour utiliser la mémoire supérieure**, il faut avoir physiquement de la RAM au delà des 640 Ko, pouvoir localiser des plages d'adresses libres (par EMM386 qui gère les UMB), et pouvoir reloger des fichiers dans ces "trous" (DEVICEHIGH et LH). Ainsi libéré, l'espace DOS libre n'est plus limité à 500 Ko, mais peut atteindre 620 Ko.

UTILISATION DE LA MEMOIRE ETENDUE

Des systèmes tels que OS/2 ou Windows NT travaillent en mode protégé et peuvent adresser directement la mémoire étendue. Avec MS/DOS, les applications doivent employer un DOS-Extender pour dépasser la limite de 1 Mo. Le **DOS Extender** permet de faire fonctionner un 286/386 en **mode protégé** (et donc accéder à la **mémoire étendue**) tout en continuant à utiliser les fonctions de MS/DOS. Il fournit donc une interface entre les applications s'exécutant en mode protégé et le DOS fonctionnant en mode réel.

FONCTION DE RECOPIE DE ROM EN RAM (mémoire fantôme, shadow RAM, RAM reflet)

Le procédé appelé "**mémoire fantôme**" consiste à recopier le code écrit en ROM (lente !) dans une RAM (rapide !), et à utiliser les possibilités de gestion de mémoire virtuelle pour placer cette RAM dans la zone d'adressage de la ROM.

Par exemple, dans certaines machines, les primitives du BIOS ou de la ROM vidéo sont automatiquement chargées en mémoire fantôme à la mise sous tension.



EXEMPLES D'UTILITAIRES DE GESTION MEMOIRE

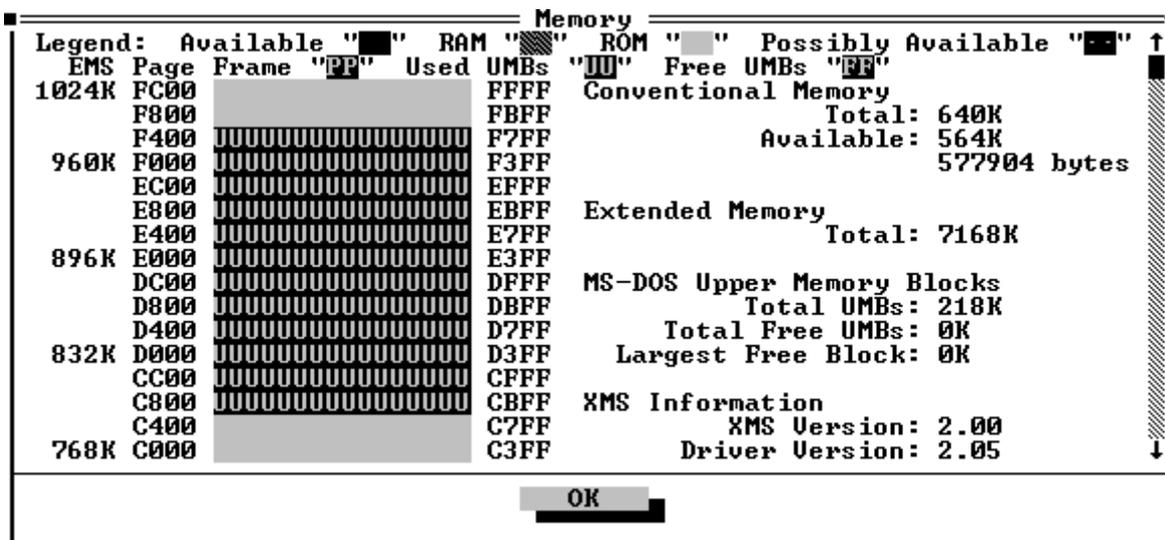
☞ **MEM** de MS/DOS : affiche les quantités de mémoire utilisée et libre.

Modules utilisant la mémoire sous 1 Mo :						
Nom	Totale		=	Conventionnelle +		Supérieure
-----	-----	-----		-----	-----	-----
MSDOS	22 061	(22K)		22 061	(22K)	0 (0K)
HIMEM	1 168	(1K)		1 168	(1K)	0 (0K)
EMM386	3 120	(3K)		3 120	(3K)	0 (0K)
etc...						
MOUSE	17 088	(17K)		0	(0K)	17 088 (17K)
SMARTDRV	27 536	(27K)		0	(0K)	27 536 (27K)
KEYB	6 944	(7K)		0	(0K)	6 944 (7K)
etc...						
Libre	549 104	(536K)		549 104	(536K)	0 (0K)

Etat de la mémoire :					
Type de mémoire	Taille	=	Utilisée	+	Libre
-----	-----		-----		-----
Conventionnelle	655 360		106 256		549 104
Supérieure	162 544		162 544		0
Réservé	131 072		131 072		0
Etendue (XMS)	7 439 632		6 391 056		1 048 576
-----	-----		-----		-----
Mémoire totale	8 388 608		6 790 928		1 597 680
Total sous 1 Mo	817 904		268 800		549 104
Taille maximale du programme exécutable			549 088	(536K)	
Bloc maximal de mémoire supérieure libre			0	(0K)	
MS-DOS résident en mémoire haute (HMA).					

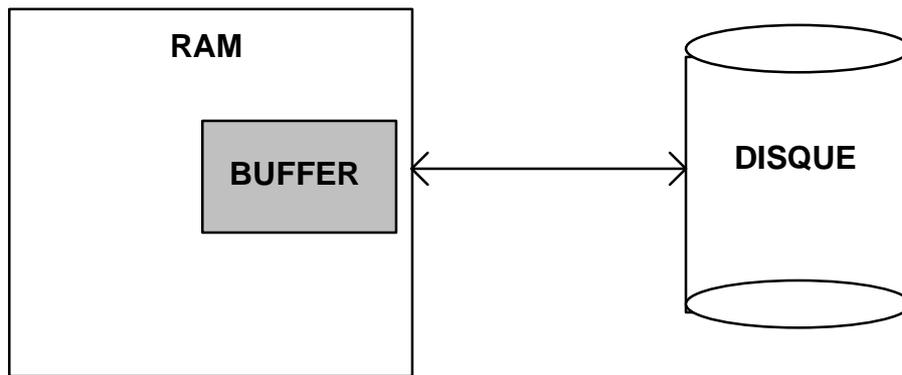
☞ **MEMMAKER** de MS/DOS : optimise automatiquement la mémoire en jouant sur le chargement des résidents et des drivers en mémoire supérieure.

☞ **MSD** des logiciels Microsoft : cartographie de la mémoire supérieure, liste des résidents et des drivers.



☞ **MFT** de Quaterdeck

OPTIMISATION MEMOIRE : BUFFER



BUFFER (mémoire tampon)

☞ Un **buffer** est une zone mémoire tampon qui peut être placée dans la **mémoire centrale** ou être liée à une **interface de périphérique** (contrôleur disque, imprimante, ...). En général un buffer fonctionne suivant le principe **FIFO** (First In First Out), c'est-à-dire que les données sont gérées comme une file d'attente où l'on rentre d'un côté et l'on sort de l'autre ; les données les plus anciennes sont donc effacées en priorité.

☞ Les **buffers d'entrée/sortie disque** (en mémoire centrale) augmentent la vitesse d'exécution des logiciels. En effet, le processeur stocke en RAM non seulement le segment appelé, mais aussi les segments contigus. Le fait de posséder en mémoire centrale des morceaux importants du ou des fichiers en cours d'utilisation permet de diminuer le temps d'accès des lectures (par contre, pour l'écriture, les données sont toujours écrites directement sur le disque en plus de figurer dans le buffer). Dans la pratique, il faut trouver un compromis entre :

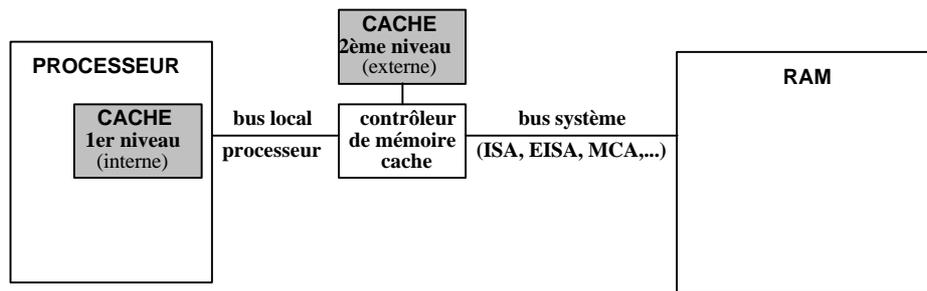
- pas assez de buffers, ce qui pénalise le débit du disque,
- et trop de buffers, ce qui encombre la mémoire (certains fichiers volumineux peuvent refuser de se charger).

GESTION DES BUFFERS AVEC MS/DOS (commande **BUFFERS** dans le **Config.sys**)

Pour **MS/DOS**, les **buffers d'entrée/sortie disque** sont attribués automatiquement par le système à chaque fichier ouvert ou rajouté par la configuration. Le système en utilise aussi pour les répertoires, sous-répertoires et pour la FAT.

MS/DOS conserve dans cette mémoire tampon les dernières données lues sur disque. Lors de la lecture suivante, il commence par cette zone de mémoire, si les données recherchées s'y trouvent, il évite un nouvel accès au disque (le gain de temps pour les lectures disque est de 10% à 15%).

OPTIMISATION MEMOIRE : CACHE PROCESSEUR



MEMOIRE CACHE PROCESSEUR (antémémoire)

Le cache a été inventé au début des **années 60** pour les **gros ordinateurs**. Un **cache processeur** est constitué de **SRAM** (Static Random Access Memory), c'est-à-dire de **RAM statique** plus rapide (de 10 à 25 ns) que les composants traditionnels (mais beaucoup plus chère, ce qui limite leur utilisation).

Le cache accélère les échanges entre le processeur et la RAM en palliant le temps d'attente du processeur (wait state). Le taux de réussite des caches (hit rate) peut s'élever jusqu'à 96 % pour les systèmes les plus performants (adressage associatif à plusieurs voies en Write back synchrone). Ceci veut dire que seulement 4% des opérations impliquent un trafic sur le bus. Le bus se trouve donc souvent libre pour permettre des lecture/écriture mémoire indépendantes du processeur (telles qu'avec DMA).

CACHE DE PREMIER ET DE DEUXIEME NIVEAUX

Le cache peut être de **2ème niveau** (cache externe apparu sur les PC avec le 386) ou de **1er niveau** (caches internes des processeurs 486 et Pentium), les deux ne s'excluant pas.

- ☞ Le **cache de 1er niveau** est très limité en taille, mais particulièrement rapide car il fonctionne à la fréquence du processeur (cache **interne** au processeur). Le i486 contient 8 Ko de cache interne, et le Pentium et le Pentium Pro contiennent 2 x 8 Ko.
- ☞ Le **cache de 2ème niveau** est en général externe au processeur. Sur les PC, la taille du cache est en général de 256 Ko, 512 Ko ou 1 Mo (serveur). Cependant, il faut remarquer que le Pentium Pro possède un cache de 2ème niveau interne au processeur de 256 ou 512 Ko suivant les versions.

DIFFERENTES ARCHITECTURES DE CACHE

- ☞ Le cache peut être exclusivement employé pour les **instructions**, ou pour les **données**. Cependant, il peut aussi être "**unifié**", c'est-à-dire contenir à la fois des instructions et des données. Les nouveaux processeurs, tels que Pentium et Power PC, intègrent des caches distincts pour les instructions et les données.
- ☞ Le cache peut être **synchrone** ou **asynchrone**. Le cache synchrone (**burst cache**, ou cache en rafale), c'est-à-dire réglé sur l'horloge du processeur, est le plus performant.

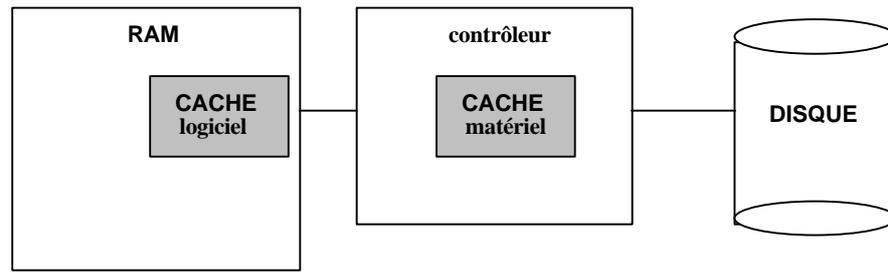
☞ Différentes **procédures de lecture** peuvent être utilisées :

- **Look through** : le **contrôleur fonctionne en série** entre processeur et RAM. Le **taux d'utilisation** du bus mémoire est **réduit** au minimum avec seulement des accès à la RAM lorsqu'il y a échec en cache. Mais les **accès RAM** sont **ralentis** puisqu'ils passent chaque fois par le contrôleur.
- **Look aside** : le **contrôleur fonctionne en parallèle** du bus mémoire, tous les accès sont simultanément doublés vers la RAM et le cache. Le **taux d'utilisation** du bus mémoire est **élevé**, ce qui pénalise les accès à la RAM des composants périphériques (vidéo,...). Mais les **accès RAM** du processeur sont **accélérés** puisque le contrôleur ne les intercepte pas. Avec cette technique, il est possible de faire des cartes mère avec cache optionnel.

☞ Différentes **procédures d'écriture** peuvent être utilisées :

- **Write through** (écriture immédiate) : le cache ne fonctionne qu'en lecture. Dès que les données sont écrites dans le cache, elles sont recopiées dans la RAM. Cette technique se trouve dans le cache interne du 486 (cache de 1er niveau) et dans les caches de 2ème niveau des machines d'entrée de gamme (mais tend à disparaître).
- **Posted write** (écriture retardée) : l'écriture en mémoire n'est effectuée que lorsque le cache a accès au bus. Cette technique, plus performante que l'écriture immédiate et plus sûre que l'écriture différée, se trouve dans les caches de 2ème niveau des machines de milieu de gamme (mais elle ne convient pas pour les grands volumes de données).
- **Write back** (écriture différée) : le cache fonctionne également en écriture. On écrit en mémoire lorsque la place occupée dans le cache par cette information doit être libérée. Cette technique est la plus performante. Cette technique se trouve dans les caches internes de données du Pentium et du 68040, ainsi que dans les caches de 2ème niveau des machines haut de gamme.

OPTIMISATION MEMOIRE : CACHE DISQUE



CACHE DISQUE

☞ Un **cache disque** peut être vu comme un **buffer géré intelligemment**.

Un **cache disque** (qu'il soit matériel ou logiciel) réduit le temps d'accès aux données en lecture (données déjà présentes dans le cache), et parfois diffère les écritures afin de rendre plus rapidement la main.

Attention, il ne faut pas prendre un cache d'une **taille trop importante**, car le temps de gestion de cette mémoire risque de dégrader les performances au lieu de les améliorer.

☞ **Fonctionnement en lecture**

Un cache conserve les **données fréquemment utilisées**. Par exemple, pour une base de données, une requête d'enregistrement peut ne demander qu'un seul accès disque, les recherches d'index s'effectuant directement en RAM (la table d'index étant souvent consultée).

Toutes les études montrent que dans 70% des cas (pour MS/DOS ou UNIX) le disque exécute une lecture séquentielle ; il lit les données placées sur les segments du disque, les uns derrière les autres. Un cache peut donc **anticiper les lectures** en chargeant quelques secteurs logiques qui suivent celui qui a été demandé (read-ahead). Cette méthode trouve ses limites quand le disque est très fragmenté.

☞ **Algorithmes utilisés pour vider une partie du cache**

En général un cache effectue une **gestion statistique des secteurs logiques** avec un algorithme **LRU** (Least Recently Use) qui tient compte de la fréquence d'accès à un fichier. La sélection s'opère sur l'hypothèse qu'un fichier non lu depuis longtemps ne sera pas lu prochainement (alors que le buffer efface le secteur le plus ancien).

☞ **Fonctionnement en écriture**

Comme pour les caches processeurs, on peut avoir une **écriture immédiate** (write through) ou **différée** (write back).



En **écriture différée**, le cache conserve plusieurs ordres d'écriture afin d'écrire en une seule fois toutes les modifications qui affectent la même piste du disque (delayed write) ; mais, si l'on éteint brusquement l'ordinateur, on risque de perdre des données (bien que le cache force l'écriture des modifications à intervalle régulier).

Le cache logiciel **SMARTDRV** du **DOS 6.2** offre une option de cache en écriture qu'il est conseillé de désactiver ; en effet, ce cache attend près de 5 secondes après une commande pour écrire sur le disque. Par ailleurs, il réorganise l'écriture afin de limiter les déplacements de la tête de lecture ; par conséquent, de très sérieuses difficultés risquent d'apparaître si l'on quitte brutalement une application (données perdues, références croisées entre fichiers).

☞ Comparaison cache logiciel et matériel

L'amélioration des performances est de 40% pour un **cache logiciel** et de 75% pour un **cache matériel**.

En cas de **cohabitation des deux types de cache**, les différents tests effectués ne concordent pas, certains trouvent un gain de performances, et d'autres une diminution (car cela revient à copier et conserver deux fois des copies des données, à les mettre à jour en double,...).

CACHE DISQUE MATERIEL (disque RAM, disque séquentiel)

Un **cache disque matériel** est constitué de RAM très rapide (de 1 à 2 Mo) implantée sur le contrôleur du disque dur.

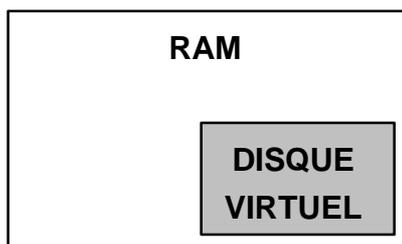
Le cache peut être segmenté ou non : en cas de cache segmenté et de système multitâche, chaque utilisateur peut se voir attribuer un segment de disque cache.

Le cache peut être sauvegardé par batterie ou non : Pour des raisons de sécurité, les données sont immédiatement réécrites sur le disque ; sauf si le contrôleur est équipé d'un cache constitué de mémoire statique C-MOS sauvegardée par batterie. Ainsi, plus besoin de réécriture physique immédiate et, en cas de coupure de courant, le cache garde en mémoire les données et les réécrit sur le disque dès que le courant est rétabli, le tout de façon transparente à l'utilisateur.

CACHE DISQUE LOGICIEL

Un cache disque logiciel est constituée d'une partie de la RAM traditionnelle gérée par un logiciel de cache. Par exemples, SmartDrv de MS/DOS et VCache de Windows 95.

OPTIMISATION MEMOIRE : DISQUE VIRTUEL



DISQUE VIRTUEL (RAMDISK)

Un **disque virtuel** est une portion de la mémoire centrale utilisée comme une unité de disque. Un tel disque est donc d'un **accès beaucoup plus rapide** que pour les disques durs. Par contre, contrairement aux buffers ou aux caches, l'utilisateur a pour charge de sauvegarder lui-même sur un véritable disque les fichiers qui y auront été entreposés. De plus, le disque virtuel diminue la taille de la RAM utilisateur.

Un disque virtuel peut être utilisé comme disque tampon destiné à héberger :

- ☞ Les **fichiers temporaires** d'un traitement quelconque (tri, indexation, répertoire TEMP du DOS et de Windows,...).
- ☞ Les **fichiers permanents** utilisés par un programme : surtout les fichiers nécessitant pas ou peu de mise-à-jour (par exemple, un dictionnaire d'un traitement de texte). Pour les fichiers modifiés, il faut prévoir un traitement qui sauvegardera le disque virtuel sur un disque classique.
- ☞ Les **applications** qui nécessitent des **accès disque fréquents** (compilateur).
- ☞ Les **applications** utilisant des fichiers de recouvrement (**overlays**).
- ☞ L'**environnement de développement** : compilateur, linker et bibliothèques.

Contre-indications : applications n'accédant que rarement au disque, ou protégées contre la copie, et environnement sensibles aux coupures de courant.

GESTION DES DISQUES VIRTUELS AVEC MS/DOS

- ☞ La **portion mémoire considérée comme une unité de disque est située** dans les 640 Ko de base, dans la mémoire XMS ou dans la mémoire EMS.
- ☞ **Gestion du disque virtuel** : déclaration de **RAMDRIVE.SYS** de Windows et MS/DOS dans le Config.sys. La commande **SET TEMP=D:** de l'Autoexec.bat permet de placer les fichiers temporaires de Windows dans un disque virtuel (ce que l'on n'envisage que si l'on a au moins 8 Mo de RAM).

ALLOCATION MEMOIRE : MEMOIRE VIRTUELLE

ALLOCATION MEMOIRE

L'**allocation mémoire** consiste à réserver de blocs mémoires nécessaires à l'exécution d'un programme.

☞ **Programme d'allocation** : pour réaliser l'allocation, le système d'exploitation doit connaître l'ensemble des blocs qui n'entrent pas (ou n'entrent plus), à un instant donné, dans la représentation d'informations en mémoire. De tels blocs sont dits "libres" et l'ensemble est appelé espace libre. Le programme d'allocation est un module du système ayant pour fonction de gérer les blocs libres et les blocs alloués.

☞ **Allocations statique et dynamique**

- **Gestion statique** : réservation de place en mémoire avant le début d'un traitement, cette place n'étant effectivement allouée que lors de l'exécution.

- **Allocation par programme utilisateur.**
- **Allocation statique par le système.**
- **Partition** : la mémoire est divisée en partitions de dimensions fixes, chaque utilisateur dispose d'une partition déterminée de mémoire.

- **Gestion dynamique** : la réservation de place en mémoire n'est effectuée qu'à l'exécution du programme selon les besoins.

- **Allocation par registre de base.**
- **Segmentation (MS/DOS).**
- **Pagination (OS/2, Windows).**

MEMOIRE VIRTUELLE (utilisée par Windows)

Le concept de **mémoire virtuelle** désigne le fait que chaque application se voit allouer un **espace d'adressage virtuelle** couvrant une capacité bien plus grande que la taille physique de la mémoire centrale.

Un gestionnaire de mémoire (**MMU : Memory Management Unit**) effectue la conversion des **adresses virtuelles** (connues par l'application) en **adresses physiques**.

Quand la **RAM** est sur le point d'être saturée, le gestionnaire de mémoire transfère des données depuis la mémoire centrale vers un **fichier d'échange** sur **disque**. Lorsque ces données sont requises par une application, elles sont réintégrées à la mémoire centrale. Ce système d'aller-retour entre mémoire centrale et disque est appelé **swapping** et est réalisé par **allocation dynamique**.

ALLOCATION MEMOIRE : SEGMENTATION

SEGMENTATION SUR UN MICRO 16 bits (8086, 80286)

- ☞ La gestion et l'accès physique à la mémoire par le microprocesseur est effectuée suivant le principe de la **segmentation**. l'adressage doit permettre d'accéder à 1 Mo.
- ☞ **Accès à 64 Ko** : Le micro ne dispose que de **registres de 16 bits** (d'où son nom) ce qui ne lui permet d'**adresser** en théorie que **64 Ko** ($2^{16} = 65536$). Cette zone contiguë est un **segment** (sa taille ne peut excéder 64 Ko).
- ☞ **Registre de base et offset.**

Pour accéder à 1 mot mémoire, on utilise 2 quantités de 16 bits :

- **1 registre de segment (RS) = registre de base**
- **1 registre déplacement = Offset**

On définit ainsi un système de numération positionnel, le **RS** représente les dizaines (en hexa) et l'**Offset** les unités.

$$\boxed{\text{adresse physique} = \text{RS} \times 16 + \text{déplacement}} \quad (16=10h)$$

- ☞ **Accès à 256 Ko et à 1 Mo.**

Le microprocesseur possède **4 registres de bases**, donc **4 segments**, ce qui veut dire qu'à un instant T, il ne peut accéder qu'à $4 \times 64 \text{ Ko} = \mathbf{256 \text{ Ko}}$ maximum. C'est en changeant la valeur de ces registres qu'il pourra accéder à **1 Mo** de la mémoire.

Les segments sont indépendants et peuvent être adjacents ou non, disjoints, partiellement recouverts ou confondus.

- ☞ Sur un **8086** : **20 bits d'adresse** permettent d'accéder à $2^{20} = 1\,048\,576$ (**1 Mo**) allocations différentes.
- ☞ Sur un **80286** : **24 bits d'adresse** permettent d'accéder à $2^{24} = 16\,777\,216$ (**16 Mo**) allocations différentes.

EXEMPLE DE CALCUL D'ADRESSE : RS=45FEh Offset=01A3h

$$\text{Adresse physique} = 10h \times 45FEh + 01A3h = 45FE0h + 01A3h = \mathbf{46183h}$$

L'adresse physique peut donc se représenter suivant deux méthodes :

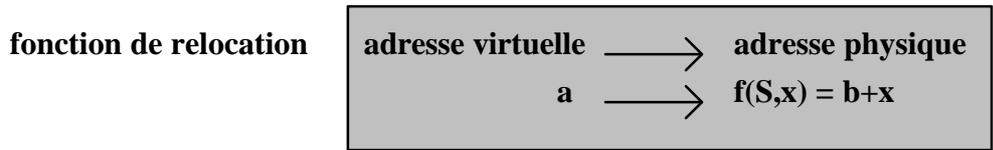
- ☞ Soit le **registre de base et le déplacement**, par exemple : **45FE : 01A3**.
- ☞ Soit la **valeur absolue** sur 5 caractères hexa, par exemple : **46183h**.

MEMOIRE VIRTUELLE PAR SEGMENTATION

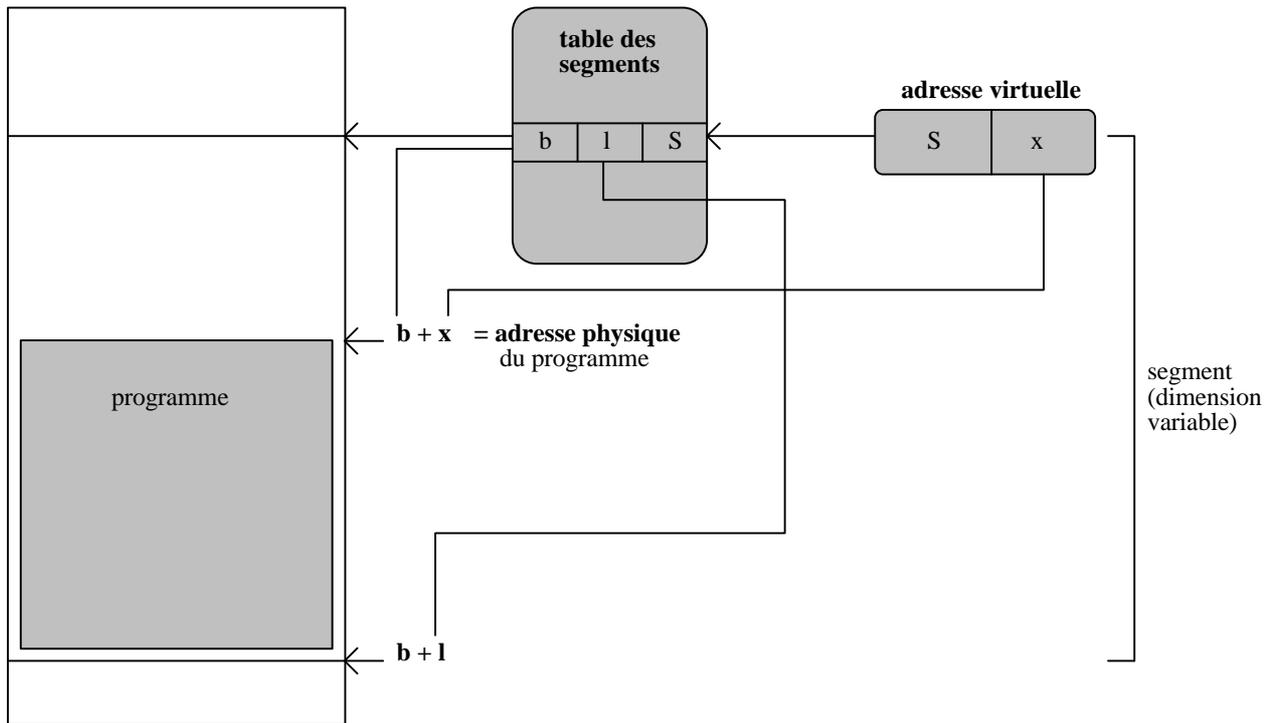
L'espace virtuel est découpé en segments de longueur quelconque qui correspondent chacun à un module de programme ou à un ensemble de données. Les différentes zones logiques d'un programme (code, données et pile) peuvent se loger dans des segments différents

L'espace adresse virtuel est à 2 dimensions, une adresse virtuelle de programme se compose de 2 parties :
• n° de segment
• n° du mot dans le segment

La table des segments (descripteurs de segments) établit la correspondance :



- a adresse virtuelle composée de 2 valeurs : S et x
- S n° de segment
- x adresse dans le segment
- b adresse de base du segment
- l longueur du segment



Un segment doit pouvoir être partagé par plusieurs programmes à la fois, cet accès est contrôlé par un système de protection.

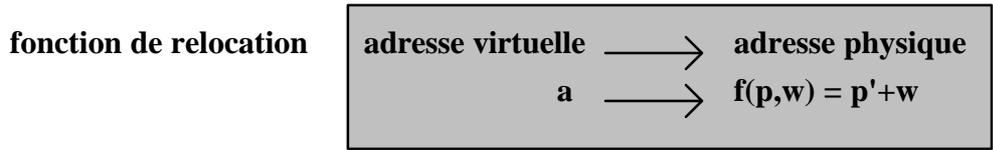
Windows 3.1 gère la mémoire virtuelle sur la base de la **segmentation**. Le module **Win386.exe** découpe la mémoire en segments de 64 Ko adressés par des adresses 16 bits.

ALLOCATION MEMOIRE : PAGINATION

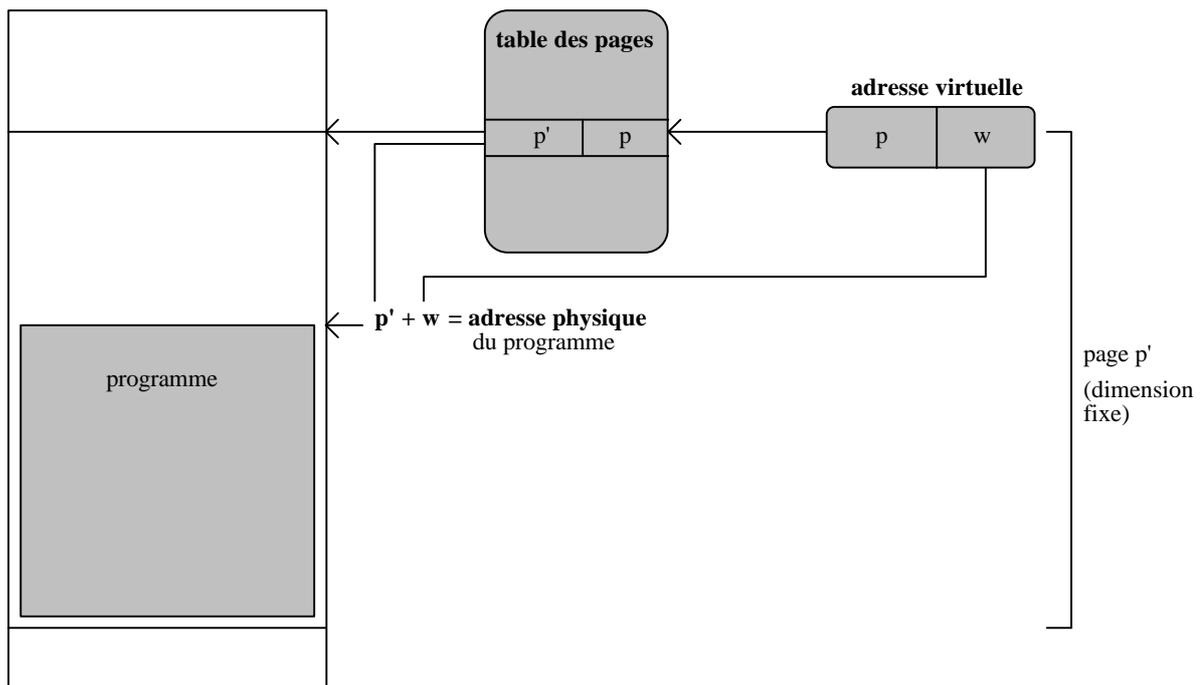
MEMOIRE VIRTUELLE PAR PAGINATION

L'espace d'adressage virtuel (adresses utilisées par les programmes) est **divisé en pages d'égales dimensions**.

A un instant donné, un processus particulier ne possède qu'un faible nombre de pages résidant en mémoire centrale (pages actives), alors que les autres résident en mémoire secondaire (pages inactives). Le système gère une **table des pages** (table de relocation déterminant la fonction de relocation f) qui permet de reconstituer l'ordre logique des différentes parties des programmes.



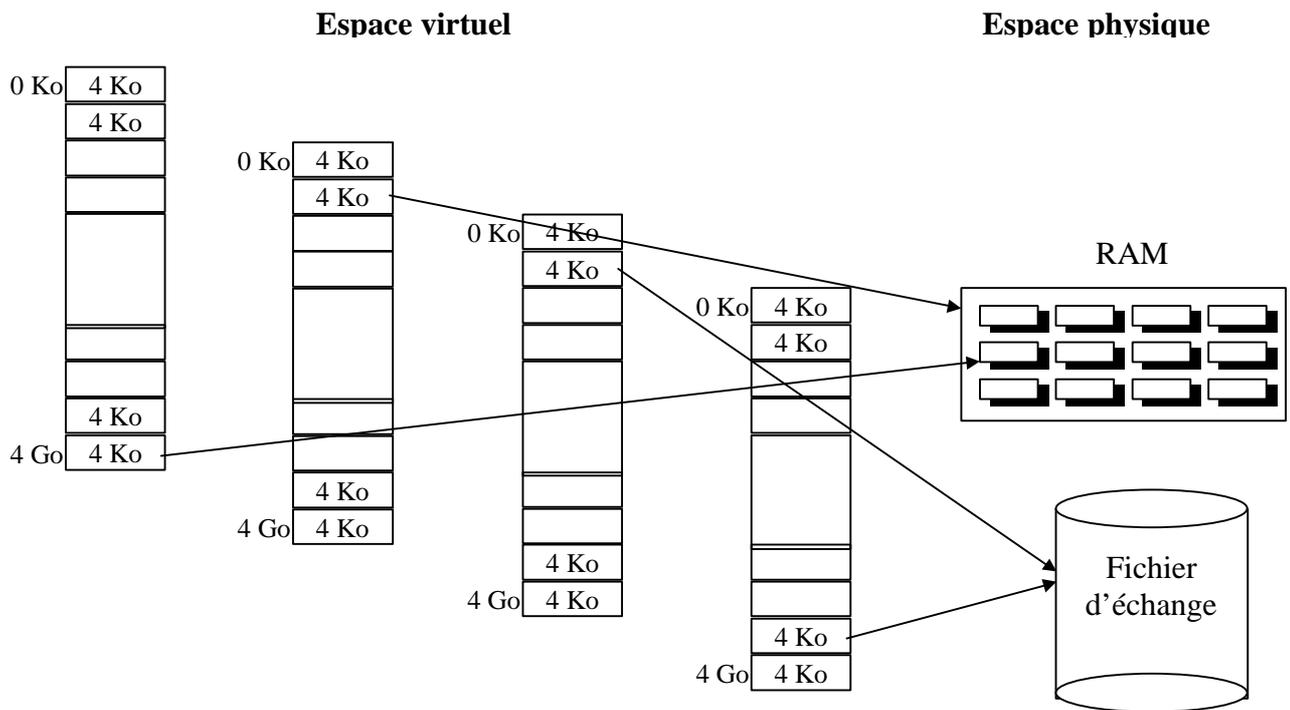
- a** adresse virtuelle composée de 2 valeurs : p et w
- p** n° de page virtuelle
- w** n° de mot dans la page (virtuelle et physique)
- p'** n° de page physique



MEMOIRE VIRTUELLE DE WINDOWS 95

☞ **Windows 95** gère la mémoire virtuelle par la technique de **pagination** qui manipule le code et les données regroupés en pages de 4 Ko ; une page peut correspondre à de la mémoire située réellement en RAM ou dans le fichier d'échange (sur disque).

Chaque **application 32 bits** et **MS-DOS** se voit attribuer un espace d'adressage distinct. Par contre, toutes les **applications 16 bits** partagent le même espace virtuel.



C'est le **gestionnaire de mémoire virtuelle VMM** (Virtual Memory Manager) qui se charge de la pagination en gérant un **tableau des pages** existant dans la mémoire centrale. Ce tableau indique les pages transférées, les processus propriétaires,... Les applications accèdent à la mémoire via les **adresses virtuelles**, qui sont mappées sur des **adresses physiques** par le VMM. Ces adresses physiques peuvent pointer sur des emplacements en **RAM** ou sur **disque** (dans le fichier d'échange).

Le mécanisme de mémoire virtuelle de Windows 95 exploite la fonctionnalité de pagination à la demande des processeurs Intel (à partir du 80386).

☞ **Les adresses virtuelles :**

0	Applications MS/DOS et Windows 16 bits	S'il existe une machine virtuelle MS/DOS , ces adresses sont utilisées par les applications DOS. Dans le cas contraire, ces adresses ne sont pas utilisées.
1 Mo	Applications Windows 16 bits	Normalement pas utilisé. Windows NT se charge au-dessus de cette adresse. Pour des raisons de compatibilité, Windows 95 n'utilise pas cet espace, de même que les applications 32 bits. Cependant, cet espace peut être exploité par certaines applications 16 bits.
4 Mo	Applications Windows 16 bits et 32 bits	Utilisé par les applications Windows 32 bits (et certaines applications 16 bits).
1.5 Go	DLLs et autres objets partagés	Utilisé par les DLLs et d'autres objets partagés. Par exemple, les boîtes de dialogue standards sont regroupées dans COMMDLG.DLL).
2 Go	Kernel, GDI, logiciels de l'anneau 0	Réservé au système d'exploitation (pilotes virtuels,...).
4 Go		

EXERCICES

Consigne de réalisation des exercices : cocher la case de chaque exercice quand celui-ci est réalisé.

Utiliser l'**aide en ligne de MS/DOS** pour avoir la syntaxe exacte des commandes, ainsi que des informations complémentaires.

MATERIEL

Visualisation des informations du Setup.

CONFIGURATION

Effectuer la cartographie de la mémoire centrale (faire des recopies d'écran afin de garder une trace papier de l'occupation mémoire) :

Occupation mémoire : conventionnelle, supérieure, HMA, XMS, EMS.

Classement des résidents et drivers selon l'utilisation de la mémoire (taille, localisation,...).

Utilisation mémoire du système : table des vecteurs d'interruptions, zone de données du Bios, ressources (FILES, BUFFERS,...), fichiers ouverts, environnement, mémoire vidéo.

Faire passer un résident de la mémoire conventionnelle à la mémoire supérieure.

Décharger un résident réseau et le recharger en interactif.

Exécuter la commande MEMMAKER et vérifier son résultat sur les fichiers de configuration.

TESTS

Créer un disque virtuel et l'utiliser pour copier des fichiers, les fichiers temporaires,...

TRAVAIL A RENDRE

☞ Informations du Bios.

☞ Cartographie de la mémoire après optimisation.



afpa ©	auteur	centre		formation	module	séq/item	type doc	millésime	page 38
	M. ASTRE	Beaumont					sup. form.	05/97 - v2.0	MEMOIRE