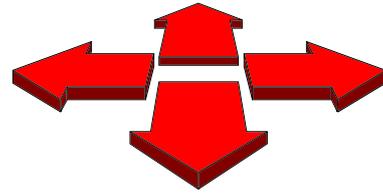




TMSI
TMSI
TMSI

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

Organisation logicielle d'un PC Interruptions DOS et BIOS

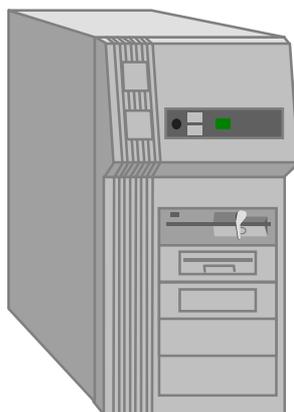


P.COLLET
09.97
version 3.0

SOMMAIRE



| | | |
|-----------|--|-----------|
| 1 | ORGANISATION DE LA MÉMOIRE D'UN PC SOUS MSDOS..... | 3 |
| 2 | QU'EST-CE QUI LIMITE L'UTILISATION DE LA MÉMOIRE ?..... | 3 |
| 3 | ORGANISATION DES 640 K OCTETS DE LA RAM (SOUS MSDOS). | 4 |
| 3.1 | TABLE DES VECTEURS D'INTERRUPTIONS..... | 5 |
| 3.2 | LA TABLE DES VARIABLES DU BIOS..... | 6 |
| 3.3 | ENVIRONNEMENT :..... | 6 |
| 3.4 | COMMAND.COM (INTERPRÉTEUR DE COMMANDE DE MS/DOS)..... | 6 |
| 3.5 | RESSOURCES SYSTÈME..... | 7 |
| 3.6 | PROGRAMMES RÉSIDENTS (TSR : TERMINATE AND STAY RESIDENT)..... | 7 |
| 3.7 | GESTIONNAIRE DE PÉRIPHÉRIQUE (DIT « DRIVER » OU « PILOTE »)..... | 8 |
| 3.8 | ORGANISATION DES 384 DERNIERS K OCTETS DE LA MEMORY MAP. | 9 |
| 4 | ORGANISATION DES ENTRÉES/SORTIES..... | 11 |
| 5 | UTILISATION DES INTERRUPTIONS LOGICIELLES. | 12 |
| 6 | LES DIFFÉRENTES COUCHES LOGICIELLES DU PC | 13 |
| 7 | INTERACTION ENTRE L'ÉLECTRONIQUE, LE BIOS ET LE DOS. | 13 |
| 8 | ANNEXE : LES VARIABLES BIOS..... | 15 |
| 9 | ANNEXE : ORGANISATION EN COUCHES | 16 |
| 10 | ANNEXE : ACQUISITION D'UNE TOUCHE AU CLAVIER..... | 17 |
| 11 | MANIPULATIONS AVEC DEBUG..... | 19 |
| 12 | UTILISATION D'UTILITAIRES..... | 20 |



1 Organisation de la mémoire d'un PC sous MSDOS

Le MS/DOS a été conçu pour les processeurs 8086/8088 qui ne pouvaient adresser **qu'1 Mo** de mémoire (20 fils d'adresses) dont 640 K de RAM. Les processeurs actuels peuvent adresser 4 Go de mémoire (32 fils d'adresses) *ce qui dans une première approche ne change rien car le MSDOS les fait fonctionner en mode réel*. Dans ce mode ces microprocesseurs se comportent comme des 8086, bien sûr plus rapide.

☞ 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.

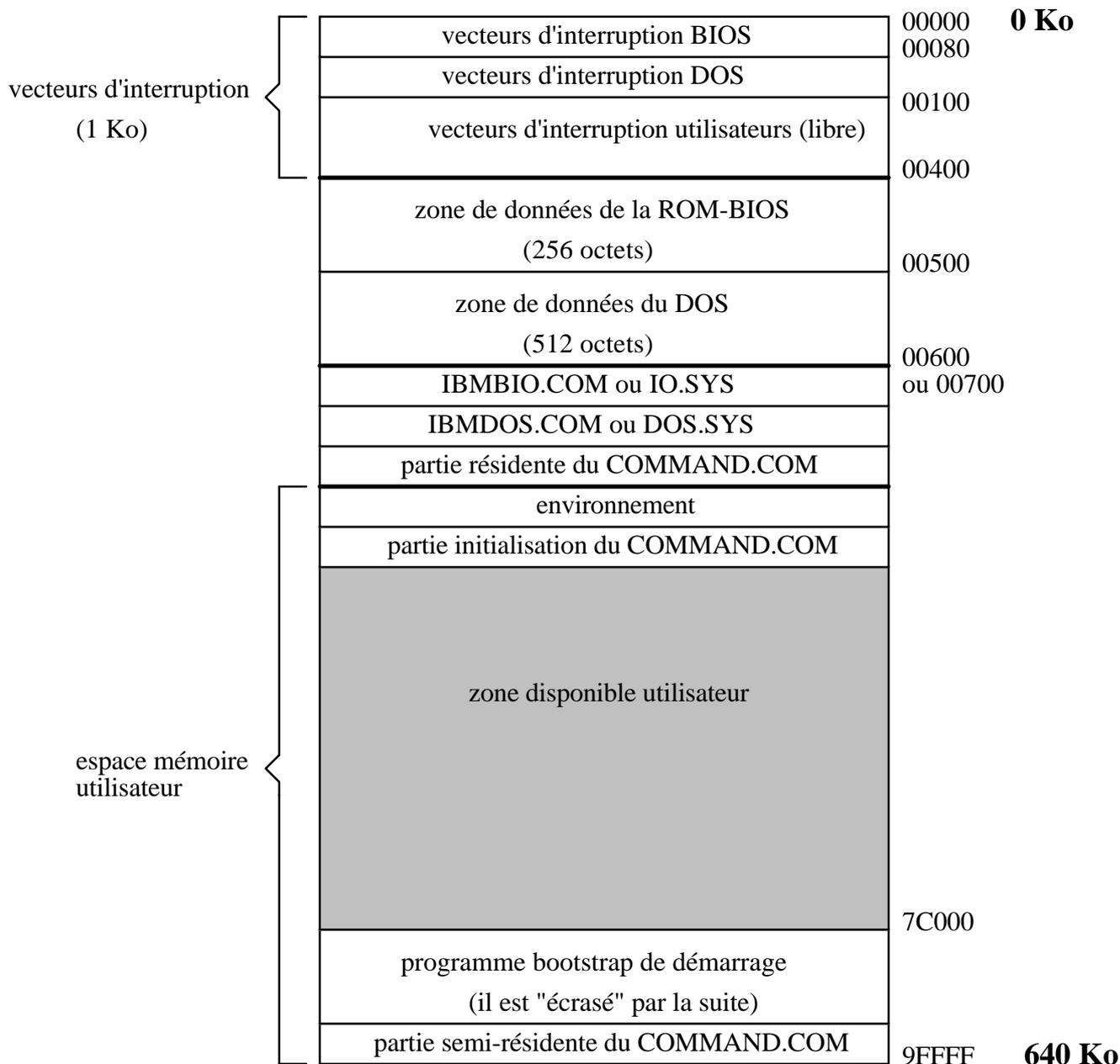
Microsoft a introduit avec la version 5 de MSDOS des fonctionnalités de gestion de la mémoire qui permettent de s'affranchir de ces limites, voir le cours sur la gestion mémoire sous MSDOS 6 pour plus d'informations.

2 Qu'est-ce qui limite l'utilisation de la mémoire ?

- ☞ **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)

3 Organisation des 640 K octets de la RAM (sous MSDOS).

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



* Les adresses basses de la RAM de **00000 à 006FF** sont réservées par le système. Il y stocke des informations nécessaires à son bon fonctionnement :

- Les vecteurs d'interruptions généraux (Hardware).
- Les vecteurs d'interruptions du BIOS.
- Les vecteurs d'interruptions du DOS.
- Enfin des tables de données nécessaires aux différents programmes du BIOS du DOS.

3.1 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 le traitement du programme en cours afin de s'occuper de cette requête. En fonction du **N° de l'interruption**, il déclenche un **programme (gestionnaire d'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**".

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

| | | | |
|----|----|----|----|
| E8 | 4E | 9A | 01 |
|----|----|----|----|

 se lit 019A:4EE8

☞ Il y a 3 types d'adresse, celles qui "pointent" sur la :

- ROM-BIOS ('F' en début de segment).
- Mémoire RAM (sous-programme du DOS ou programme résident <drivers>).
- 0, si l'interruption n'est pas gérée.

| Int | Address | Owner | Usage/Comments |
|-------|-----------|----------|---------------------------------|
| 00 | 0116:108A | DOS | CPU: Divide Error |
| 01 | 0070:06F4 | IO | *CPU: Single Step / Deb.Except. |
| 02 | 04E6:0016 | STACKS | NMI (Non Maskable Interrupt) |
| 03 | 0070:06F4 | IO | *CPU: Breakpoint |
| 04 | 0070:06F4 | IO | *CPU: INTO detected Overflow |
| 05 | F000:FF54 | ROM BIOS | PrtSc / CPU: BOUND exceeded |
| 06 | F000:8008 | ROM BIOS | CPU: Invalid opcode |
| 07 | F000:EF6F | ROM BIOS | CPU: Proc.Ext. not available |
| 08 | DB41:1875 | SMARTDRV | IRQ0:Timer /CPU:double except. |
| 09 | DB41:1923 | SMARTDRV | IRQ1:Keyb. /CPU:PE prot.error |
| 0A | F000:EF6F | ROM BIOS | IRQ2:8259#2/CPU:Invalid TSS |
| 0B | F000:EF6F | ROM BIOS | IRQ3:COM2 /CPU:Seg.not present |
| 0C | F000:EF6F | ROM BIOS | IRQ4:COM1 /CPU:Stack Fault |
| 0D | F000:EF6F | ROM BIOS | IRQ5:LPT2 /CPU:GPF |
| 0E | 04E6:00B7 | STACKS | IRQ6:FD /CPU:Page Fault |
| 0F | 0070:06F4 | IO | *IRQ7:LPT1 /CPU:Coproc. Error |
| 10 | DB41:18B3 | SMARTDRV | Video-BIOS functions |
| 11 | F000:F84D | ROM BIOS | BIOS: get cfg./CPU:Alignm.Chk. |
| 12 | F000:F841 | ROM BIOS | BIOS: get RAM size |
| 13 | DB41:18C5 | SMARTDRV | BIOS: floppy/hard disk func. |
| 14 | F000:E739 | ROM BIOS | BIOS: serial communic.interf. |
| 15 | DB41:19A0 | SMARTDRV | BIOS: extended functions |
| ----- | | | |
| 6F | F000:EF6F | ROM BIOS | free |
| 70 | 04E6:0052 | STACKS | IRQ8: CMOS Real-Time clock |
| 71 | F000:ECF3 | ROM BIOS | IRQ9: redirected IRQ2 |
| 72 | F000:EF6F | ROM BIOS | IRQ10: reserved |
| 73 | F000:EF6F | ROM BIOS | IRQ11: reserved |
| 74 | D2F3:14DD | MOUSE | IRQ12: resvd./ pointing device |
| 75 | F000:F0FC | ROM BIOS | IRQ13: math coprocessor |
| 76 | 04E6:0117 | STACKS | IRQ14: hard disk controller |
| 77 | 04E6:012F | STACKS | IRQ15: reserved |

Figure 1 : extrait de la table des vecteurs d'interruptions d'un PC

Voir en annexe les différentes fonctions des interruptions logicielles.



3.2 La table des variables du BIOS.

Cette zone comprise entre 0040:0000 et 0050:000 est utilisée par le BIOS pour le stockage de ses variables internes; voir en annexe la définitions des différentes cellules mémoires pour un PC standard.

| Offs. | Value | Description |
|-------|--------|-------------------------------------|
| 00h | 03F8h | Base I/O addr.of 1st serial port |
| 02h | 02F8h | Base I/O addr.of 2nd serial port |
| 04h | 0000h | Base I/O addr.of 3rd serial port |
| 06h | 0000h | Base I/O addr.of 4th serial port |
| 08h | 0378h | Base I/O addr.of 1st parallel port |
| 0Ah | 0000h | Base I/O addr.of 2nd parallel port |
| 0Ch | 0000h | Base I/O addr.of 3rd parallel port |
| 0Eh | 035Fh | 4th parallel port / [PS] Ext.BDS |
| 10h | 4427h | Installed hardware: |
| | b15-14 | 1 number of parallel devices |
| | b13 | 0 [Conv, PS/2-55LS] Internal modem |
| | b12 | 0 reserved |
| | b11-9 | 2 number of serial devices |
| | b8 | 0 reserved |
| | b7-6 | 0 number of diskette drives minus 1 |
| | b5-4 | 10 Initial video mode: 80x25 Farbe |
| | b3 | 0 reserved |
| | b2 | 1 [PS] pointing device |
| | b1 | 1 =1 if math co-processor |
| | b0 | 1 =1 if diskette available for boot |
| 12h | 00h | POST status |
| 13h | 640 | Base memory size in k (0-640) |
| 15h | 0 | [PC,XT] Adapater memory size in k |

Figure 2 : Extrait de la zone BIOS (fourni par AGSI)

3.3 Environnement :

Zone de mémoire qui contient les variables d'environnement du DOS (PATH, PROMPT, COMSPEC, variables définies avec SET).

3.4 COMMAND.COM (interpréteur de commande de MS/DOS)

Ce fichier est visible, c'est lui qui réalise l'interface entre l'utilisateur et le système. Il contient en particulier les commandes DOS résidentes, les messages d'erreurs. Il est organisé en trois modules:

- ☞ Une **partie résidente** qui contient les modules de traitement des interruptions logicielles 22h, 23h et 24h, programme de rechargement de la partie semi-résidente, gestions d'erreurs et messages d'erreur, commandes internes, gestion CTRL+C. Ce sont ces routines qui affiche les messages de type:

Abort, Retry, Ignore

- ☞ Une **partie transitoire** qui contient par exemple les routines pour sortir les messages d'interrogation (A>) et qui gère les entrées clavier de l'utilisateur. Elle contient aussi les commandes internes telles DIR, COPY, etc... Ce module n'est pas protégé et peut donc être effacé par un programme d'application. A la fin de l'exécution d'un programme d'application la partie résidente du processeur de commande détermine si la partie transitoire a été ou non effacée par le programme d'application et la recharge en RAM s'il y a lieu.

- ☞ Un **module d'initialisation** qui est chargé du lors du lancement du système et dont le rôle est d'initialiser le système. Une fois le travail d'initialisation terminé la zone mémoire qu'il occupe est libérée.

3.5 Ressources système.

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)

3.6 Programmes résidents (TSR : Terminate and Stay Resident)

Un programme résident est un programme qui est chargé en mémoire dans une zone "protégée"¹ et qui est peut être activé par différents événement : par une interruption (matérielles ou logicielles), par une combinaison de touches pour certains utilitaires (SIDEKICK,...).

Normalement, lorsqu'un programme se termine, MSDOS ferme les fichiers que ce programme aurait oublié de fermer, puis récupère la mémoire occupé par celui-ci (c'est la fonction 4C de l'interruption 21 qui réalise ce travail). Dans le cas d'un résident, on utilise une autre fonction pour retourner au DOS qui ne libère pas la mémoire.

☞ Programmes résidents systèmes :

• 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** qui 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 "Gestion mémoire").

¹ En réalité on peut toujours écrire dans ces zones par programme.

3.7 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) |

MSDOS fournit des drivers qui le cas échéant peuvent être remplacés, il permet de plus l'ajout de nouveaux drivers. 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, carte son ...

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.

3.8 Organisation des 384 derniers K octets de la memory MAP.

| 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 |
| | | <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">MEMOIRE SUPERIEURE</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">mémoire vidéo</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">espace disponible pour des extensions</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">réservé</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">ROM-Bios</div> | A0000 | 640 Ko | segment A |
| B0000 | 704 | | segment B | | |
| C0000 | 768 | | segment C | | |
| D0000 | 832 | | segment D | | |
| E0000 | 896 | | segment E | | |
| F0000 | 960 | | segment F | | |
| F3FFF | | | | | |
| F4000 | 976 | | | | |
| F6000 | 984 | | | | |
| ROM sur la carte mère | 48 Ko | | FE000 | 1016 | |
| | | | FFFFFF | 1 Mo | |

* De 8² Ko à **64³ Ko voir 128 Ko pour les BIOS Plug and Play** selon le modèle (PC - XT - AT) et le type de BIOS sont réservés à la **ROM BIOS⁴** dans le dernier segment compris entre F0000 et FFFFF ou E0000 et FFFFF.

Remarque :

Sur un AT cette mémoire est aussi accessible dans le segment compris entre FFOOOO et FFFFFFF afin de garder une compatibilité entre le mode protégé et le mode réel.

² Pour les XT.

³ Pour les AT dont le BIOS n'est pas Plug and Play.

⁴ Basic Input Output System.

La tâche principale du **BIOS** est de fournir un certain nombre de routines⁵ d'entrée/sortie de base pour permettre la communication entre l'unité centrale et les périphériques tels le clavier, les lecteurs de disquettes, l'écran, etc.

Ces routines sont accessibles par le programmeur par l'appel d'interruptions logicielles, de 10H à 1FH.

Tout concepteur de système d'exploitation doit impérativement utiliser ces routines.

Ces différentes fonctions assurent, de ce fait, une interface standard entre le système d'exploitation et le matériel (hardware) qui n'est pas à 100 % identique d'un compatible à l'autre.

Les routines du BIOS, comme tout programme, doivent pouvoir créer, stocker, modifier des variables, une zones mémoire comprise entre 00400H (ou 0040 :0000) et 004FFH est réservée à cet effet.

La deuxième tâche du BIOS consiste à lancer le système lors de la mise sous tension.

Le microprocesseur, lors de la mise sous tension ou lors d'un Reset, exécute le programme qui commence à l'adresse FFFF0 (F000:FFFF). A cette adresse une instruction exécute un saut à une routine dont l'adresse peut varier d'un PC à l'autre, elle réalise par contre sensiblement les mêmes tâches:

- Tester les différentes fonctions du microprocesseur; ses registres et quelques instructions sont testées.
- Tester la ROM BIOS.
- Tester les différents circuits de la carte mère, tels le contrôleur de DMA, le contrôleur d'interruption, les RAM, etc.
- Initialiser la table des vecteurs d'interruption ainsi que les variables du BIOS.
- Tester les périphériques, clavier, lecteur de disquette, etc.
- Rechercher des éventuelles extensions ROM, qui contiennent les programmes de gestion de périphériques tels : le contrôleur SCSI, la carte vidéo (EGA-VGA), une carte réseau, etc. Une ROM d'extension doit toujours commencer par les codes 55h et AAH. Le BIOS "appelle" ces routines par un CALL FAR, ce qui permet au programme contenu dans cette ROM de d'initialiser des vecteurs d'interruption ou de rediriger des vecteurs d'interruptions vers ses propres routines.
- Charger le système d'exploitation. C'est le rôle de l'interruption 19H , appelée lorsque l'on actionne simultanément les touches Alt, Ctrl, Suppr.
Cette routine charge un secteur en RAM qui doit être situé à un emplacement bien précis de la disquette ou du disque dur. Ce secteur contient une routine qui à son tour chargera le reste du système d'exploitation.

* Les adresses comprises entre **C0000** et **FFFFFF** ne sont **pas utilisées en standard**, les constructeurs de cartes d'extension peuvent les utiliser. Par exemple, la **ROM des cartes SCSI** est située dans cet espace et la **ROM de la carte SVGA** entre **C0000H** et **C7FFFH**. Ces ROM qui sont des extensions du BIOS⁶ sont reconnues automatiquement par le programme d'initialisation de la ROM BIOS.

* Les adresses comprises entre **A0000** et **BFFFF** sont réservées à la mémoire des différentes **cartes vidéo**. Par exemple les adresses **A0000** à **AFFFF** sont réservées à la **RAM vidéo des cartes EGA, VGA et SVGA en mode graphique**, de même que les adresses comprises entre **B8000** et **BFFFF** sont utilisées par ces **cartes en mode texte**



⁵ Programmes initialisant et gérant les divers circuits d'interface.

⁶ Elles contiennent des routines spécialisées dans la gestion du matériel de la carte d'extension et fournissent des interruptions logicielles qui sont utilisées par les programmeurs.

Remarque :

Pour mémoire la carte CGA utilise les adresses B8000 à BFFF et la **carte MDA monochrome** utilise les adresses comprises entre **B0000 et B0FFF**.

4 Organisation des entrées/sorties.

| Circuit | PC_XT | PC_AT |
|--------------------------------------|---------|---------|
| Contrôleur DMA (8237) | 000-01F | 000-01F |
| Contrôleur d'interruption (8259) | 020-03F | 020-03F |
| Temporisateur | 040-05F | 040-05F |
| Interface parallèle programmable | 060-06F | - |
| Monocontrôleur clavier 8042 | - | 060-06F |
| Horloge temps réel MC1468118 | - | 070-07F |
| Registre de page DMA | 080-09F | 080-09F |
| Registre de masque NMI | 0A0-0BF | - |
| Contrôleur d'interruption N°2 (8259) | - | 0A0-0BF |
| Contrôleur DMA N°2 (8237) | - | 0C0-0DF |
| Coprocasseur mathématique | - | 0F0-0F1 |
| Coprocasseur mathématique | - | 0F8-0FF |
| Contrôleur de disque dur | 320-32F | 1F0-1F8 |
| Adaptateur de jeux (joysticks) | 200-207 | 200-207 |
| Unité d'extension | 210-217 | - |
| Deuxième port parallèle | | 278-27F |
| Deuxième port série | 2F8-2FF | 2F8-27F |
| Premier port parallèle | 378-37F | 378-37F |
| Carte d'écran monochrome | 3B0-3BF | 3B0-3BF |
| Carte vidéo CGA | 3D0-3DF | 3D0-3DF |
| Contrôleur de disquette | 3F0-3F7 | 3F0-3F7 |
| Premier port série | 3F8-3FF | 3F8-3FF |

Le 8086 et ses successeurs peuvent adresser 64 K octets d'entrées/sorties. Dans l'IBM PC seules les adresses comprises entre 0000 et 03FF sont utilisées. Les micro actuels qui utilisent un bus PCI ne sont plus limités à 3FF

5 UTILISATION DES INTERRUPTIONS LOGICIELLES.

Les ressources du DOS et du BIOS sont accessibles par le programmeur par l'appel d'interruptions logicielles. Pour gérer les entrées/sorties telles que clavier, écran, imprimante, disque, le programmeur appelle une interruption, après avoir chargé les registres du microprocesseur avec des valeurs adaptées.

Exemple :

```

MOV AH 01 ; lecture d'un caractère au clavier
INT 21H   ; avec écho sur l'écran.
          ; le code ASCII du caractère sera récupéré dans AL
MOV AH 05 ; sortie du caractère contenu dans
INT 21H   ; AL vers l'imprimante.

```

La programmation en assembleur d'un système PC se résume le plus souvent de la façon suivante :

- 1 - Identification de la fonction désirée (voir en annexe les interruptions BIOS et DOS).
- 2 - Initialisation des différents registres du microprocesseur (paramètres d'entrées).
- 3 - Appel de l'interruption avec INT suivie du numéro de l'interruption.
- 4 - Eventuellement traitement des paramètres de retour (qui sont présents dans les différents registres du microprocesseur).

Exemples :

Pour obtenir le code ASCII et l'attribut d'affichage du caractère qui se trouve à la position courante du curseur, il faut utiliser l'interruption BIOS 10H, fonction 8.

- * Paramètres d'entrées :
 - AH = N° fonction
 - BH = N° de page
- * Paramètres de sorties :
 - AH = Attribut
 - AL = Code ASCII
- * Programme :


```

MOV AH,8
MOV BH,0
INT 10H

```

Pour afficher un caractère à l'écran, on peut utiliser la fonction 2 de l'interruption DOS 21H.

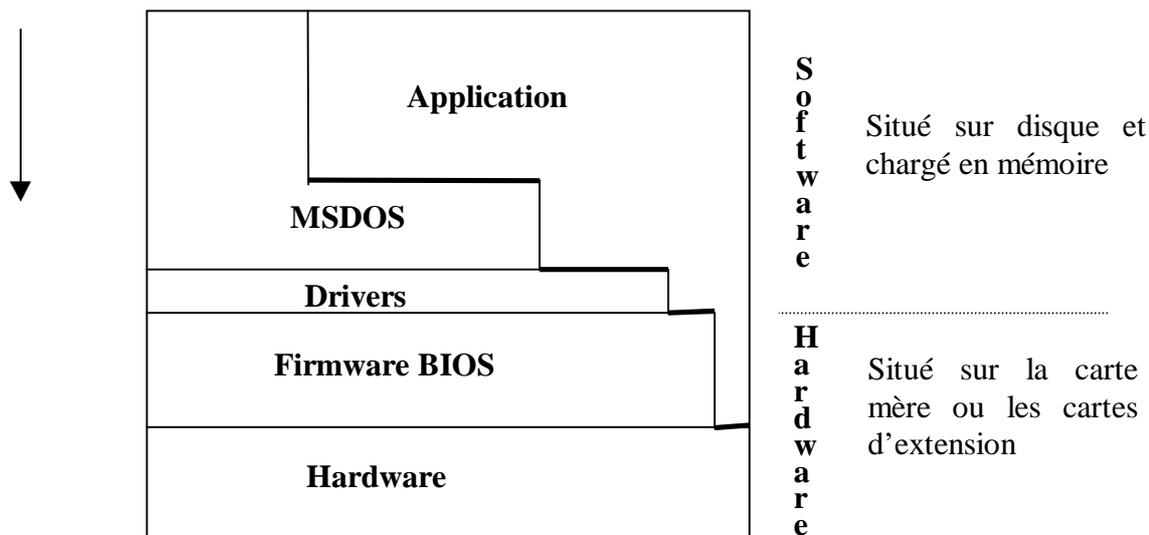
- * Paramètres d'entrées :
 - AH = N° fonction
 - DL = Code ASCII
- * Aucun paramètre de sortie.
- * Programme :


```

MOV AH,2
MOV DL,'A'
INT 21H

```

6 Les différentes couches logicielles du PC



L'utilisateur ne "voit" du micro-ordinateur que le système d'exploitation ou le programme d'application.

7 Interaction entre l'électronique, le BIOS et le DOS.

Pour bien comprendre le rôle et les interactions entre les différentes couches, prenons l'exemple de la gestion du clavier (voir en annexe les illustrations).

On peut distinguer trois niveaux dans la structure d'un PC, le DOS et le BIOS qui constituent l'aspect logiciel et le niveau le plus bas l'électronique (cartes mères, cartes d'extension et les périphériques).

La gestion du clavier est un bon exemple pour illustrer les interactions de ces différentes couches.

Le clavier est géré par un micro-contrôleur⁷. Le 8048 ou 8042 dont le rôle est de détecter si une touche est enfoncée ou relâchée. Si une touche est maintenue enfoncée plus d'une demi-seconde, le 8048 se comportera comme si la touche avait été sélectionnée dix fois au cours de la seconde.

Cette surveillance se poursuit jusqu'à ce que la touche soit relâchée. Le processeur du clavier n'associe pas de caractères ni même de code ASCII aux différentes touches. Il traite par exemple les touches SHIFT et CTRL comme les autres. Par contre il associe un code (SCAN CODE) à chaque touche.

Lorsqu'une touche est actionnée le processeur du clavier transmet le code clavier (SCAN CODE) à l'unité centrale sous la forme d'un octet, lorsque la touche est relâchée, ce même code est à nouveau transmis mais après addition de la valeur 128 (soit bit 7 à 1).

Une interruption (IRQ 1) est alors générée via le 8259 vers le microprocesseur de la carte mère, ce qui déclenche l'interruption matérielle 09 qui correspond à une routine du DOS (KEYB).

Cette routine utilise une routine du BIOS pour aller lire le SCAN CODE dans le 8255 port A (60H), puis de le convertir en un code ASCII.

Cette tâche est relativement complexe car elle doit prendre en compte les touches SHIFT, CTRL, ALT, ALT GR... Selon la combinaison des touches actionnées, le code à produire peut être soit un code ASCII soit un code dit "clavier étendu" : par exemple les flèches, le retour chariot.

Une fois le code déterminé il est placé dans le buffer clavier du BIOS dont la taille est 16 octets.

⁷ Microprocesseur qui intègre des fonctions d'entrées/sorties.

Si le buffer est plein la routine émettra un bip indiquant un débordement du buffer clavier. Le travail de l'interruption 09 est alors achevé et le processeur peut poursuivre son travail.

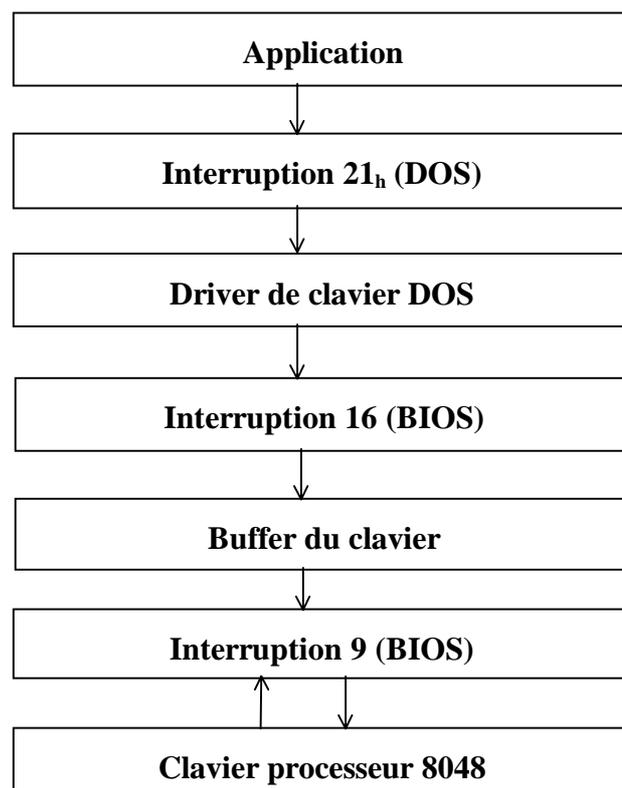
Pour lire les caractères entrés dans le buffer, afin de l'afficher par exemple, il faut utiliser une interruption Software du BIOS. Il s'agit de l'interruption 16 du BIOS qui recouvre 3 fonctions.

Ces 3 fonctions peuvent être appelées par un programme utilisateur, ou par le système d'exploitation grâce à l'instruction INT 16H en ayant mis dans AH, le numéro de la fonction.

Le niveau supérieur est le DOS, celui-ci offre différentes routines qui permettent de gérer le clavier ; ces routines sont regroupées sous le nom "driver clavier". Leur fonction consiste à lire des caractères au clavier à l'aide des fonctions BIOS de l'interruption 16H pour stocker dans un buffer ou bien, pour certaines, à vider le buffer clavier du BIOS.

On peut substituer le driver clavier du DOS par un driver plus puissant appelé ANSI.SYS, ceci en l'indiquant dans config.sys par la directive DRIVER = ANSI.SYS, ce qui permet par exemple de programmer les touches de fonction.

Les fonctions du driver clavier sont accessibles grâce à l'interruption Software DOS INT 21H avec le code de la fonction dans AH. Ces fonctions sont plus complexes que celles du BIOS, par exemple on peut obtenir le code ASCII du caractère issu du clavier et l'afficher sur l'écran. Ces diverses fonctions sont appelées par les programmes d'applications.



L'avantage de cette structure, qui est lourde, c'est que tout programme d'application qui n'utilise que des appels aux interruptions DOS peut fonctionner sur des ordinateurs dont l'électronique est différente mais qui travaillent sous MS-DOS.

Un autre avantage c'est d'offrir au programmeur un ensemble de fonctions "toutes prêtes".

Enfin on peut modifier le matériel à condition de réactualiser le BIOS sans pour autant devoir modifier MS-DOS et encore moins les programmes d'applications.

8 ANNEXE : Les variables BIOS

00 - 07 : Adresse des ports d'E/S série.

ex: 0040:0000 F8 03 F8 02 00 00 00 00

08 - 0F : Adresse des ports parallèles.

ex: 0040:0007 78 03 00 00 00 00 00 00

10 - 11 : Configuration de la machine. L'interruption 11h lit ces octets.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

- Bit 0 : 1 s'il y a au moins 1 lecteur de disquette.
- Bit 1 : 1 s'il y a un coprocesseur.
- Bit 2-3 : taille de la RAM pas utilisée sur AT.
- Bit 4-5 : carte vidéo.
 - 00 - adaptateur spécial.
 - 01 - adaptateur couleur.
 - 10 - adaptateur couleur.
 - 11 - adaptateur monochrome.
- Bit 6-7 : nombre de lecteur de disquettes
 - 00 - 1 lecteur.
 - 01 - 2 lecteurs.
 - 10 - 3 lecteurs.
 - 11 - 4 lecteurs.
- Bit 8 : inutilisé.
- Bit 9-A-B : nombre de ports série.
- Bit C : 1 si joystick installé
- Bit D : inutilisé.
- Bit E-F : 1 si touche insert enfoncé.

13 - 14 : taille de la RAM. Utilisé par l'interruption 12h.

17 : octet d'état du clavier

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

- Bit 4 : 1 si Scrool Lock activée.
- Bit 5 : 1 si Num Lock activé.
- Bit 6 : 1 si Caps Lock activé.
- Bit 7 : 1 si Insert activé.



APPLICATION

Attente et lecture sur le périphérique d'entrée courant (clavier par défaut) d'un caractère, puis affichage du caractère sur le périphérique de sortie courant (écran) et actualisation de la position courante du curseur.

```
MOV AH,01
INT 21H
```

MSDOS (Routine correspondant à INT21 fonction 01)

```
MOV AH,00
INT 16H
```

BIOS (Routine qui lit un caractère dans le buffer clavier. Si aucun caractère n'est présent, elle attend la frappe d'une touche. Le caractère est retourné dans AL).

IRET

```
MOV AH,0AH
INT 10H
```

BIOS (Routine qui affiche le caractère contenu dans AL à la position courante du curseur).

IRET

```
MOV AH,02
INT 10H
```

BIOS (Routine qui positionne le curseur à la place suivante).

IRET

IRET

```
MOV AH,4CH
INT 21H
```

11 Manipulations avec Debug

1 - Recherche de l'adresse de début du BIOS

visualiser le début du BIOS en déduire la version et le nom du fabricant

Note : Selon le type de PC (XT ou AT) la taille de la ROM BIOS varie de 16 à 64 Ko.

2 - "Désassembler" les premières instructions exécutées lors de la mise sous tension.

Note : Utiliser le cours pour retrouver la façon dont le microprocesseur exécute un RESET.

3 - Identifier le type de PC

Note : Cette information est codée à l'adresse physique FFFFE

FE = XT ou PC portable

FC = AT

FF = PC

4 - Rechercher les différentes ROM

<< S >>

note : Les ROM optionnelles contenues sur les cartes d'extension commencent toutes par 55AA.

En déduire la "memory map" de votre PC.

5 - Recherche et "désassemblage" d'une routine d'interruption.

exemple INT 09

Note : voir le cours pour calculer l'adresse du vecteur d'interruption.

exemple INT 16

6 - Examiner la zone d'adresses comprises entre B800:0 et B800:7D0

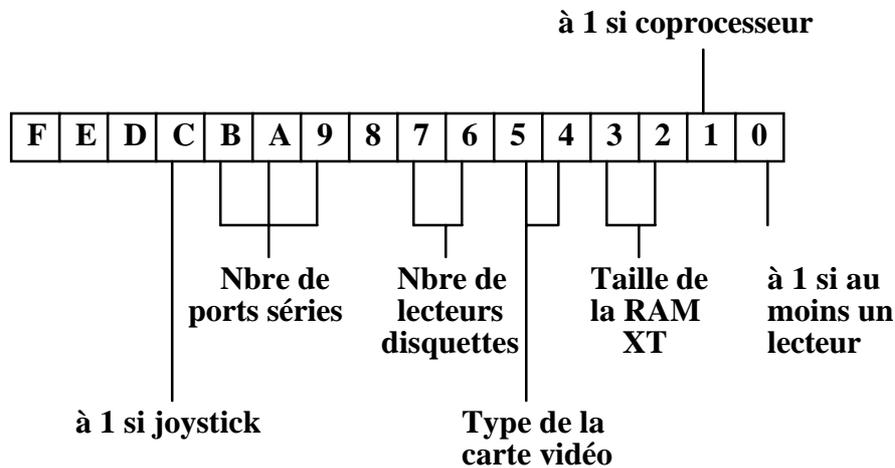
Modifier par exemple l'adresse B800:400 et B800:401

Remarque ? (regarder l'écran)

7 - Examen de la zone des variables du BIOS

adresse physique 400 à 4FF

- * 0 à 7 adresses des ports séries
COM1 - COM2 - COM3 - COM4
- * 8 à F adresses des ports parallèles.
LPT1 - LPT2 - LPT3 - LPT4
- * 10 à 11 configuration



Pour plus d'information voir la bible PC page 438

8 - Modification d'une donnée d'un programme.

Exemple : Mettre une <, > à la place du <., > sur le pavé numérique du clavier

```
Debug keyb.com
D 670
E 676 ==> 2E '!' mettre 2C ',' ou pour certain Keyb.com E 6A6
W
```

12 Utilisation d'utilitaires

Etablir la Memory map, déterminer l'occupation de la mémoire conventionnelle et faire un rapport contenant toutes les informations techniques (type de processeur, type de carte vidéo, type de BIOS, etc.) de votre ordinateur en utilisant les utilitaires suivants:

- CHECKIT (menu info système).
- MSD (MSDOS).
- SYSINFO (NORTON).
- SYSCHK (SHAREWARE).
- SYSINFO (AMIDIAG)
- AGSI
- PC-CONFIG

