

## Chapitre 2

# Spécifications des interruptions du système d'exploitation

Nous avons vu, lors de notre étude de la programmation en langage machine ou d'assemblage en utilisant les appels système (les interruptions logicielles) du système MS-DOS, les spécifications de ceux-ci. Cependant nous avons pris en exemple les appels système des versions évoluées de MS-DOS par rapport à MS-DOS 1.25.

Puisque notre but est d'en voir l'implémentation pour MS-DOS 1.25, il faut donner les spécifications des appels système de cette version.

La référence est le manuel d'utilisation, distribué avec MS-DOS 1.25 : les 46 fonctions de l'interruption 21h y sont commentées dans l'appendice D.

## 2.1 Les entrées-sorties standard

Les entrées-sorties standard n'ayant pas évoluées, du point de vue de leur spécification, elles sont conformes à ce que nous en avons dit. Rappelons-les donc très rapidement.

Fonction 01h de saisie d'un caractère avec écho à l'écran.- La fonction 01h de l'interruption 21h permet de saisir un caractère frappé au clavier, de le placer dans le registre AL et de l'afficher à l'écran.

Fonction 02h d'affichage d'un caractère.- La fonction 02h de l'interruption 21h permet d'afficher à l'écran le caractère dont le code ASCII se trouve dans le registre DL.

Fonction 03h de lecture d'un caractère sur l'interface auxiliaire.- La fonction 03h de l'interruption 21h permet de lire un caractère sur l'interface de communication asynchrone. Elle y attend un caractère et le place dans le registre AL.

Il n'y a pas de tampon (contrairement au clavier) et n'est pas pilotée par interruption.

Au démarrage, MS-DOS 1.25 initialise le premier port auxiliaire à 2 400 bauds, sans parité, avec un bit d'arrêt et un mot de 8 bits.

Les fonctions 03h et 04h ne renvoient pas de code de statut ou d'erreur. Pour un meilleur contrôle, il vaut mieux utiliser l'interruption 14h du BIOS.

Fonction 04h d'envoi d'un caractère sur l'interface auxiliaire.- La fonction 04h de l'interruption 21h permet d'envoyer le caractère contenu dans le registre DL à l'interface de communication asynchrone.

Fonction 05h d'impression d'un caractère.- La fonction 05h de l'interruption 21h permet d'imprimer le caractère dont le code ASCII est situé dans le registre DL sur la première imprimante LPT1.

Fonction 06h d'entrée-sortie directe sur la console.- La fonction 06h de l'interruption 21h permet une entrée-sortie directe sur la console :

- Si le contenu du registre DL est FFh :
  - si tampon du clavier n'est pas vide, le drapeau de zéro est baissé et le premier caractère du tampon est placé dans AL ;
  - si le tampon est vide, le drapeau de zéro est levé.
- Si le contenu de DL n'est pas FFh, le registre DL est censé contenir un caractère valide, qui doit être affiché. Cette fonction ne teste pas la combinaison de touches Ctrl-Break.

Fonction 07h d'entrée directe sur la console sans écho.- La fonction 07h de l'interruption 21h permet une entrée directe à partir de la console sans écho à l'écran. Elle attend qu'un caractère se trouve dans le tampon du clavier et renvoie le code ASCII de celui-ci dans le registre AL. Comme pour la fonction 06h, elle ne teste pas la combinaison de touches Ctrl-Break.

Fonction 08h de saisie d'un caractère sans écho à l'écran.- La fonction 08h de l'interruption 21h permet de saisir un caractère frappé au clavier et de le placer dans le registre AL, mais sans l'afficher à l'écran, contrairement à la fonction 01h.

Fonction 09h d'affichage d'une chaîne de caractères.- La fonction 09h de l'interruption 21h affiche la chaîne de caractères d'adresse DS:DX. Celle-ci doit se terminer par le caractère '\$' (de code ASCII 24h).

Fonction 0Ah de saisie dans un tampon.- La fonction 10 de l'interruption 21h permet de saisir des caractères au clavier et de les placer dans un tampon.

Avant d'appeler cette fonction, le couple de registres DS:DX doit pointer sur un tampon. Le premier octet de celui-ci, non nul, spécifie le nombre maximum de caractères que peut contenir le tampon.

Elle saisit les caractères au clavier et les place dans le tampon en commençant par les placer au troisième octet du tampon. La lecture au clavier et le remplissage du tampon sont effectués jusqu'à ce qu'on appuie sur la touche 'Enter'. Lorsqu'on arrive à l'avant dernier caractère que le tampon peut contenir, chaque caractère additionnel est ignoré et cause l'émission d'un bip, jusqu'à ce qu'on appuie sur la touche 'Enter'.

Le second octet du tampon contient le nombre de caractères reçus, sans compter le caractère retour chariot (de code ASCII 0Dh), qui est toujours le dernier caractère.

Fonction 0Bh de vérification du statut.- La fonction 11 de l'interruption 21h vérifie le statut du clavier.

Si un caractère est disponible, AL renvoie FFh et 00h sinon.

Fonction 0Ch de vidage du tampon du clavier.- La fonction 12 de l'interruption 21h vide le tampon du clavier puis exécute la fonction dont le numéro est dans AL (seuls 01h, 06h, 07h, 08h et Ah sont permis). Ceci force le système à attendre qu'on appuie sur un caractère.

## 2.2 Gestion de la date et de l'heure

Fonction 2Ah de récupération de la date.- La fonction 42 de l'interruption 21h renvoie la date dans les registres AL, CX et DX, avec :

- CX contenant l'année (en binaire dans l'amplitude 1980–2099),
- DH le mois (1 pour janvier, 2 pour février, etc.),
- DL le jour du mois (dans l'amplitude 1–31),
- AL le jour de la semaine (0 pour dimanche, 1 pour lundi, etc.).

Fonction 2Bh d'initialisation de la date.- La fonction 43 de l'interruption 21h initialise la date, celle-ci étant donnée par les contenus des registres AL, CX et DX, avec :

- CX contenant l'année (en binaire dans l'amplitude 1980–2099),
- DH le mois (1 pour janvier, 2 pour février, etc.),
- DL le jour du mois (dans l'amplitude 1–31),
- AL le jour de la semaine (0 pour dimanche, 1 pour lundi, etc.).

Si la date n'est pas valide AL renvoie FFh, sinon 00h.

Fonction 2Ch de récupération de l'heure.- La fonction 44 de l'interruption 21h renvoie l'heure dans le couple de registres CX:DX, avec :

- CH contenant l'heure (0–23),
- CL les minutes (0–59),
- DH les secondes (0–59),
- DL les 1/100 ièmes de secondes (0–99).

Fonction 2Dh d'initialisation de l'heure.- La fonction 45 de l'interruption 21h initialise l'heure, celle-ci étant donnée par le contenu du couple de registres CX:DX :

- CH contenant l'heure (0–23),
- CL les minutes (0–59),
- DH les secondes (0–59),
- DL les 1/100 ièmes de secondes (0–99).

Si l'heure n'est pas valide AL renvoie FFh, sinon 00h.

## 2.3 Les acteurs du système de fichiers

### 2.3.1 Le lecteur de disquette par défaut

Puisque le PC d'IBM dispose, dans sa deuxième version, de deux lecteurs de disquette, l'un est considéré *par défaut*, c'est-à-dire que c'est de lui dont on parle lorsqu'on ne spécifie rien de particulier. Le lecteur de disquette A a pour numéro 1 et B pour numéro 2.

### 2.3.2 La zone de transfert (DTA)

Le tampon du disque, ou **zone de transfert** (DTA pour *Disk Transfer Area*), est la zone de mémoire vive utilisée pour contenir les données issues d'une lecture sur la disquette, ou à transférer lors d'une écriture :

- On peut la placer à n'importe quel emplacement de la mémoire vive.
- On ne peut choisir qu'une seule DTA à un moment donné. Une fois choisie, MS-DOS continue à l'utiliser pour toutes les opérations sur la disquette jusqu'à ce qu'une autre zone soit choisie.
- Elle doit être spécifiée avant toute opération de lecture ou d'écriture sur une disquette.

### 2.3.3 Les secteurs de la disquette

Nous avons vu que l'intérêt d'un système d'exploitation est de s'abstraire du matériel en considérant la notion de *fichier*. La lecture ou l'écriture d'un secteur donné, spécifié par ses caractéristiques physiques (face, piste, secteur) peut être effectué avec les interruptions logicielles du BIOS. La première étape d'un système d'exploitation est de lire ou écrire un secteur, ou de plusieurs secteurs consécutifs au sens de la numérotation logique, de la disquette : le système d'exploitation abstrait en effet les caractéristiques physiques par une numérotation linéaire des secteurs.

La lecture ou l'écriture de secteurs consécutifs, au sens de la numérotation linéaire, est traité par IO.COM. Le noyau y revient en y ajoutant des messages d'erreur.

Organisation générale des secteurs d'une disquette.- Rappelons que MS-DOS 1.25 ne prend en compte que les disquettes simple face, d'une capacité de 160 kiO, et les disquettes double face, d'une capacité double.

#### Disquettes simple face

Une disquette simple face comprend 40 pistes (numérotées linéairement par le système d'exploitation de 0 à 39) avec huit secteurs par piste (numérotés linéairement de 1 à 8) et 512 octets par secteur.

MS-DOS alloue l'espace de la disquette de la façon suivante :

- Piste 0, secteur 1 : enregistrement d'amorçage (MBR pour *Master Boot record*), écrit par la commande FORMAT.
- Piste 0, secteurs 2-3 : deux copies de la FAT, une par secteur.
- Piste 0, secteurs 4-7 : répertoire tenant sur quatre secteurs.
- Piste 0, secteur 8 à piste 39 secteur 8 : secteurs des données.

#### Disquettes double face

Une disquette double face contient le même nombre de secteurs par piste, de même taille, mais il y en a de chacun des deux côtés.

Pour des raisons d'efficacité, MS-DOS numérote linéairement les secteurs 1-8 de la piste du premier côté puis les secteurs 1-8 du second côté, avant d'aller au premier côté de la piste suivante.

La disposition de l'espace vue à propos des disquettes simple face s'applique aux disquettes double face, sauf que le répertoire occupe trois secteurs de plus, soit 7 en tout ; les secteurs supplémentaires sont celui de la piste 0, secteur 8, côté 0, suivi de ceux la piste 0, secteurs 1-2, côté 1. L'espace des données commence donc à la piste 0, secteur 3, côté 1.

Unité d'allocation.- L'espace minimum alloué par MS-DOS pour les données sur une disquette est appelé **unité d'allocation** (*cluster* ou *unit of allocation* en anglais). Pour les disquettes simple face, l'unité d'allocation est le secteur ; pour les disquettes double face, il s'agit de deux secteurs consécutifs.

Emplacement des secteurs des fichiers.- Les fichiers ne sont pas nécessairement constitués de secteurs consécutifs, au sens de la numérotation linéaire, de la disquette. L'espace sur la disquette pour un fichier est alloué une unité d'allocation à la fois, en sautant les unités d'allocation déjà allouées. La première unité d'allocation libre est l'unité d'allocation suivante allouée, quelle que soit son emplacement sur la disquette.

Copie de MS-DOS.- Une **disquette système** contient une copie de MS-DOS. Celle-ci est placée dans la zone des données de la façon suivante pour MS-DOS 1.25 :

#### **Disquettes simple face**

- IO.COM : 4 secteurs, de la piste 0 secteur 8 à la piste 1 secteur 3.
- MSDOS.COM : 13 secteurs, de la piste 1 secteur 4 à la piste 2 secteur 8.

#### **Disquettes double face**

- IO.COM : 4 secteurs, piste 0, côté 1, secteurs 3 à 6.
- MSDOS.COM : 13 secteurs, de la piste 0, côté 1, secteurs 7-8 à la piste 1, côté 0, secteurs 1-4.

Ces deux programmes doivent résider aux emplacements indiqués pour que l'enregistrement d'amorçage, pour lequel le système de fichiers n'est pas encore initialisé, puisse les charger lors du démarrage.

### 2.3.4 Le répertoire

Il y a un répertoire unique, le *répertoire racine*, dans la version 1.25 de MS-DOS.

La commande FORMAT de MS-DOS construit le répertoire (racine, le seul pour MS-DOS 1.25) sur la piste 0, secteurs 4-7 pour les disquettes simple face, soit un total de 2 048 octets. Il y a de la place pour 64 entrées, chacune d'une taille de 32 octets. Dans le cas d'une disquette double face, le répertoire occupe sept secteurs, contenant ainsi 112 entrées.

Structure d'une entrée de répertoire.- Pour chaque fichier, MS-DOS crée une **entrée** de 32 (soit 20h) octets dans le répertoire. Celle-ci spécifie le nom du fichier, la date à laquelle il a été créé, sa taille et son unité d'allocation de départ. Voici la signification de ces 20h octets :

- 00h-07h : les huit premiers octets spécifient le nom du fichier, complété par des espaces. Le premier octet peut aussi indiquer, par une caractère non affichable, le statut du fichier :
  - + 00h : le fichier n'a jamais été utilisé.
  - + 05h : le premier caractère du fichier est en fait E5h.
  - + 2Eh : il s'agit d'un sous-répertoire (à partir de MS-DOS 2.0).
  - + E5h : le fichier a été effacé.
- 08h-0Ah : ces trois octets spécifient l'extension, par exemple EXE, complétée par des espaces.
- 0Bh : les bits 0 à 5 de l'octet 11 spécifient les attributs du fichier, conformément au format suivant (un bit à 1 indique que le fichier a la propriété indiquée) :
  - + 01h : fichier pouvant seulement être lu,
  - + 02h : **fichier caché**, c'est-à-dire qu'il ne devra pas être affiché dans le répertoire (ou sous-répertoire) à l'écran par la commande **dir**,
  - + 04h : **fichier système**, c'est-à-dire IO.COM et MSDOS.COM ; un tel fichier n'est pas non plus affiché à l'écran par la commande **dir**,
  - + 08h : **nom du volume**, celui-ci occupant l'emplacement du nom du fichier et de son extension (à partir de MS-DOS 2.0).
  - + 10h : **sous-répertoire** (à partir de MS-DOS 2.0).
  - + 20h : **fichier d'archive**. L'archivage se fait incrémentalement : lorsqu'on archive pour la première fois, tous les fichiers sont copiés et les bits d'archive sont mis à zéro ; lorsqu'on modifie un fichier, son **bit d'archive** est mis à 1 ; lorsqu'on archive la fois suivante, seuls les fichiers dont le bit d'archive est à 1 seront copiés.
- 0Ch-15h : les dix octets 12 à 21 sont réservés par le système.
- 16h-17h : ces deux octets indiquent l'heure à laquelle le fichier a été créé ou modifié pour la dernière fois. Le format en est hh/mm/xx, de la façon suivante :
 

<	hh	>	<	mm	>	<	xx	>							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

 où :
  - hh vaut 0 à 23
  - mm vaut 0 à 59
  - xx vaut 0 à 31 (par pas de 2 secondes)

- **18h-19h** : les octets 24 et 25 spécifient la date à laquelle le fichier a été créé ou a été modifié pour la dernière fois. Le format en est mm/dd/yy, de la façon suivante :

```

<      25      > <      24      >
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
y  y  y  y  y  y  y  m  m  m  m  d  d  d  d
où :

```

mm vaut 1 à 12

dd vaut 1 à 31

yy vaut 0 à 119 (depuis 1980, donc 1980 à 2099)

- **1Ah-1Bh** : les octets 26 et 27 spécifient le numéro relatif de la première unité d'allocation du fichier. La valeur 0 indique que le fichier n'a pas d'espace alloué.

Notons que la première unité d'allocation de l'espace des données porte toujours le numéro relatif 002, quel que soit le type de disquette. Pour une disquette simple face, il s'agit du secteur 8 de la piste 0. Pour une disquette double face, ce sont les deux secteurs commençant à la piste 0, secteur 3, côté 1.

Le numéro d'unité d'allocation est stocké en commençant par l'octet de plus petit poids.

- **1Ch-1Fh** : les quatre octets 28 à 31 spécifient la taille du fichier, en octets. Le premier mot spécifie la partie de poids faible de la taille. Les deux mots sont stockés en commençant par l'octet de poids faible.

Exemple.- Chargeons le répertoire racine de la première disquette de la mise à jour de MS-DOS 6.0 avec `debug` et interprétons la première donnée. Nous avons vu que ce répertoire débute au secteur 19, soit 13h.

```

- L 100 0 13 1
- D 100
17A5:0100 49 4F 20 20 20 20 20 20-53 59 53 21 00 00 00 00 IO SYS!....
17A5:0110 00 00 00 00 00 00 00 30-6A 1A 02 00 00 9F 00 00 .....0j.....
17A5:0120 4D 53 44 4F 53 20 20 20-53 59 53 21 00 00 00 00 MSDOS SYS!....
17A5:0130 00 00 00 00 00 00 00 30-6A 1A 52 00 16 95 00 00 .....0j.R.....
17A5:0140 41 50 50 4E 4F 54 45 53-54 58 54 20 00 00 00 00 APPNOTESTXT ....
17A5:0150 00 00 00 00 00 00 00 30-6A 1A 9D 00 7D 00 00 00 .....0j...}...
17A5:0160 41 55 54 4F 45 58 45 43-42 41 54 20 00 00 00 00 AUTOEXECBAT ....
17A5:0170 00 00 00 00 00 00 00 30-6A 1A 9E 00 26 00 00 00 .....0j...&...

```

Le premier fichier a pour nom `IO.SYS`. L'octet d'attribut 21h nous dit qu'il s'agit d'un fichier d'archive ne pouvant qu'être lu. Les dix octets suivants, réservés, sont tous nuls. Les deux octets suivants, 00h et 30h, nous donnent l'heure : on a 3000h, soit 0011 0000 0000 0000b; l'heure est donnée par les cinq premiers bits, soit 00110b ou 6; on a donc 6 h 00 mn 00 s.

Les deux octets suivants, 6Ah et 1Ah, nous donnent la date : on a 1A6Ah, soit 0001 1010 0110 1010b; l'année est donnée par les sept premiers bits, soit 0001101b ou 13, donc 1993; le mois est donné par les quatre bits suivants, soit 0011b, ou 3, il s'agit donc du mois de mars; le jour est donné par les cinq derniers bits, soit 01010b ou 10.

Les deux octets suivants, 02h et 00h, donnent la première unité d'allocation, soit 2.

Les quatre derniers octets, 00h, 9Fh, 00h et 00h, donnent la taille du fichier, soit 00009F00h, ou  $9 \times 4096 + 15 \times 256$ , soit encore 40 704 octets.

Lorsqu'une disquette est formatée avec l'option /S (de la commande `FORMAT`), les trois premiers fichiers du répertoire sont `IO.COM`, `MSDOS.COM` et `COMMAND.COM`.

Les deux premiers fichiers systèmes sont placés sur des secteurs spécifiques. Ils sont protégés d'un effacement accidentel en étant exclus de toute recherche sur le répertoire. Ils sont appelés *fichiers cachés* pour cette raison.



### 2.3.5 La table d'allocation des fichiers (FAT)

L'objet de la *table d'allocation des fichiers* (FAT) est de déterminer l'emplacement du fichier sur le disque. Quand on crée un nouveau fichier, ou lorsqu'on modifie un fichier existant, le système revoie les entrées de la FAT. La FAT contient une entrée pour chaque unité d'allocation du disque. Elle commence au secteur 2, suivant immédiatement le secteur d'amorçage.

Taille d'une entrée de la FAT.- Pour une disquette simple face, de capacité 160 kiO et d'unités d'allocation d'un secteur de 512 octets, on a besoin de 320 entrées (plus précisément 312 puisque les huit premiers secteurs ne sont pas occupés par des données mais réservés au système pour le secteur d'amorçage, les deux FAT et le répertoire). Une entrée, contenant un numéro d'unité d'allocation, doit donc avoir une capacité de plus d'un octet (qui permet de repérer 256 entrées). Deux octets étant beaucoup trop, les concepteurs de MS-DOS ont choisi, pour MS-DOS 1.25, une taille d'entrée de la FAT de 12 bits, soit un octet et demi (on parle de **FAT 12**).

Capacité d'adressage de la FAT.- La FAT contient une entrée pour chaque unité d'allocation du disque. Ainsi plus une unité d'allocation contient de secteurs, plus la capacité d'adressage est grande. Pour MS-DOS, la taille d'une entrée de la FAT peut être de 12 bits (cas de MS-DOS 1.25), 16 bits ou 32 bits (on parle de **FAT 12**, **FAT 16** ou de **FAT 32**) ; plus la taille est grande, plus la capacité d'adressage est grande.

Contenu d'une entrée.- Nous avons vu que la première unité d'allocation de chaque fichier est spécifiée dans le répertoire. Un fichier pouvant occuper plusieurs unités d'allocation (partiellement pour la dernière), chaque entrée de la FAT spécifie la fonction de l'unité d'allocation correspondante : réservée (c'est le cas des 8 premiers secteurs), libre, non utilisable, le numéro de l'unité d'allocation suivante pour le fichier dont fait partie cette unité d'allocation (il s'agit du cas le plus fréquent) et une valeur particulière lorsqu'il s'agit de la dernière unité d'allocation du fichier.

**Première et seconde entrées.**- La première et la seconde entrée de la FAT, correspondant à des unités d'allocation ne faisant pas parties des données, spécifie le type de disquette : FFFFFFFh pour une disquette simple face, FFFFFFFh pour une disquette double face dans le cas de MS-DOS 1.25.

**Entrées suivantes.**- Une entrée pour une FAT 12 doit être interprétée de la façon suivante :

000	unité d'allocation non utilisée mais disponible
nnn	numéro d'unité d'allocation suivante du fichier
FF0-FF6	unité d'allocation réservée
FF7	unité d'allocation non utilisable (piste endommagée)
FF8-FFF	dernière unité d'allocation du fichier

Exemple théorique en FAT 12.- Considérons l'exemple suivant :

Entree :	0	1	2	3	4	5	6	7	8	9
Valeur :	<FEF>	<FFF>	<FFF>	<004>	<005>	<009>	<FF7>	<000>	<000>	<FFF>

Les deux premières unités d'allocation, de numéro 0 et 1, contiennent la signature **FEFFFFh** pour indiquer qu'on a à faire à une disquette simple face. Le répertoire nous dit sûrement qu'un certain fichier commence à l'unité d'allocation 2; la FAT nous dit qu'il se termine également à cette unité d'allocation. Le répertoire nous dit aussi sûrement qu'un autre fichier commence à l'unité d'allocation 3; la FAT nous dit qu'il continue à l'unité d'allocation 4, puis à l'unité d'allocation 5, puis à l'unité d'allocation 9, qui est la dernière unité d'allocation de ce fichier. La FAT nous dit aussi que l'unité d'allocation 6 est endommagée et que les unités d'allocation 7 et 8 ne sont pas utilisées.

Algorithme de décodage d'une entrée de FAT 12.- Nous venons de parler d'un exemple *théorique* car le fait qu'un numéro d'unité d'allocation occupe un octet et demi, et qu'*Intel* a décidé de coder un mot sur deux octets en commençant par celui de poids faible, complique un peu le codage de l'information.

Soit  $n$  le numéro de l'unité d'allocation. On multiplie par  $3/2$  et on prend la partie entière; on obtient ainsi  $p = \text{int}(3n/2)$ ; ceci revient à multiplier par trois et à prendre le quotient exact par 2 dans la division euclidienne. On considère le mot de la FAT constitués des octets numéros  $p$  et  $p + 1$ , codant donc un nombre  $\langle p + 1 \rangle \langle p \rangle$ . Si  $n$  est impair, on a besoin des 12 bits les plus à gauche, soit les trois chiffres hexadécimaux les plus à gauche, pour obtenir le numéro de l'unité d'allocation; si  $n$  est pair, on a besoin des 12 bits les plus à droite, soit les trois chiffres hexadécimaux les plus à droite.

Considérons, par exemple, la FAT :

F0 FF FF 03 40 00 FF 0F

- Déterminons la valeur de l'entrée 2. Multiplier 2 par  $3/2$  donne 3. Les octets 3 et 4 sont 03h et 40h, ce qui donne le mot 4003h. Puisque 2 est pair, la valeur est donnée par les trois derniers chiffres hexadécimaux, soit 003h, ou 3.
- Déterminons la valeur de l'entrée 3. Multiplier 3 par  $3/2$  donne 4. Les octets 4 et 5 sont 40h et 00h, ce qui donne le mot 0040h. Puisque 3 est pair, la valeur est donnée par les trois premiers chiffres hexadécimaux, soit 004h, ou 4.
- Déterminons la valeur de l'entrée 4. Multiplier 4 par  $3/2$  donne 6. Les octets 6 et 7 sont FFh et 0Fh, ce qui donne le mot 0FFFh. Puisque 4 est pair, la valeur est donnée par les trois derniers chiffres hexadécimaux, soit FFFh, ce qui indique une fin de fichier.

Exemple.- Chargeons la FAT de la première disquette de la mise à jour du DOS 6.0 avec **debug** et interprétons les premières données. La FAT commence au secteur 1.

- L 100 0 1 1  
- D 100

Copie de la FAT en mémoire centrale.- Une copie de la FAT de la dernière disquette utilisée dans chacun des deux lecteurs de disquette est conservée en mémoire centrale.

L'octet précédent la copie de la FAT en mémoire centrale indique si le contenu de cette copie a été sauvegardée sur le disque depuis sa dernière modification en mémoire centrale (00h) ou non (FFh).

### 2.3.6 Le bloc des paramètres d'une disquette

Le **bloc des paramètres d'une disquette** (DPB pour *Disk Parameter Block*) est une structure de 20 octets spécifiant la structure d'une disquette :

- Le premier octet, champ DEVNUM, spécifie le nombre de lecteurs de disquette.
- Le second octet, champ DRVNUM, spécifie le numéro du lecteur de disquette par défaut : 1 pour A et 2 pour B.
- Les deux octets suivants, champ SECSIZ, spécifient la taille, en octets, d'un secteur.
- L'octet suivant, champ CLUSMSK, spécifie le nombre de secteurs moins un d'une unité d'allocation.
- L'octet suivant, champ CLUSSHFT, donne le logarithme en base deux du nombre de secteurs d'une unité d'allocation.
- Les deux octets suivants, champ FIRFAT spécifient le numéro du premier secteur de la FAT.
- L'octet suivant, champ FATCNT, spécifie le nombre de copies de la FAT sur la disquette.  
En fait il y en a toujours 2.
- Les deux octets suivants, champ FIRREC, spécifient le numéro du premier secteur de la première unité d'allocation.  
Il y a dû y avoir hésitation sur le nom, car on a REC pour *record* (*enregistrement*) et non CLUS pour *cluster*.
- Les deux octets suivants, champ MAXCLUS, spécifient le nombre d'unités d'allocation du disque, plus un.
- L'octet suivant, champ FATSIZ, spécifie le nombre de secteurs occupés par la FAT.  
Il s'agit toujours de 1 pour MS-DOS 1.25.
- Les deux octets suivants, champ FIRDIR, spécifient le numéro du premier secteur du répertoire.
- Les deux derniers octets, champ FAT, pointent sur le début de la copie de la FAT en mémoire centrale.

### 2.3.7 Le bloc de contrôle d'un fichier

Le bloc de contrôle d'un fichier (FCB pour *File Control Block*) contient les informations sur ce fichier (permissions, taille, etc.). Il existe des FCB *standard* pour les fichiers ordinaires et des FCB *étendus* pour la recherche d'une entrée dans le répertoire.

#### Bloc de contrôle de fichier standard

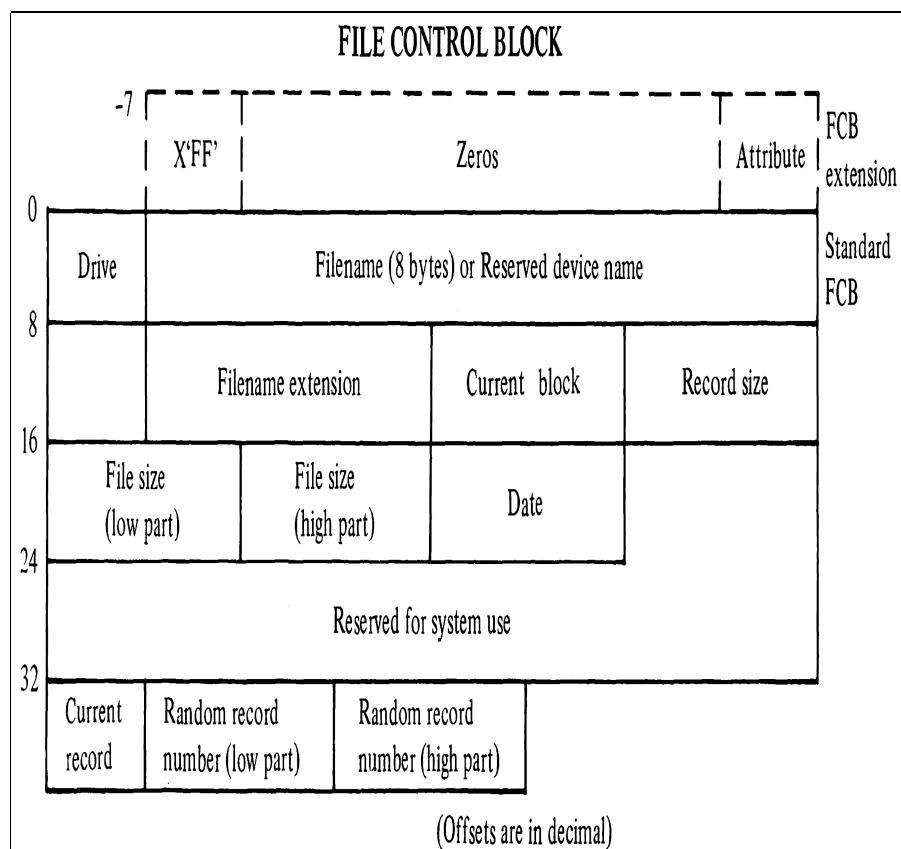


FIGURE 2.1 – Structure du FCB

La structure d'un FCB standard est la suivante, les décalages étant en décimal :

— L'octet 0 spécifie le numéro du lecteur de disquette.

Avant ouverture :

0 - lecteur par défaut (0 est remplacé par le numéro de lecteur en cours lors de l'ouverture).

1 - lecteur A

2 - lecteur B

Après ouverture :

1 - lecteur A

2 - lecteur B

— Les octets 1 à 8 spécifient le nom du fichier, justifié à gauche avec des espaces comme caractères de remplissage.

Un nom de périphérique (comme 'LPT1') ne comporte pas le caractère deux-points.

- Les octets 9 à 11 spécifient l'extension du nom de fichier, justifiée à gauche avec des espaces comme caractères de remplissage.
- Les octets 12 et 13 spécifient le numéro du bloc en cours lors d'une opération séquentielle depuis le début du fichier, la numérotation commençant à zéro. Un **bloc** est constitué de 128 enregistrements, chaque enregistrement ayant la taille spécifiée dans le champ 'taille d'enregistrement logique'.  
Le numéro de bloc est utilisé conjointement avec le champ '(numéro d')enregistrement' pour une lecture ou une écriture séquentielle.
- Les octets 14 et 15 spécifient la taille, en octets, d'un **enregistrement** (*record* en anglais). Elle est initialisée à 80h (128) par la fonction d'ouverture d'un fichier. Si ce n'est pas la taille voulue, on doit en changer la valeur à la main.
- Les octets 16 à 19 spécifient la taille du fichier, en octets. Le premier mot est celui de poids faible.
- Les octets 20 et 21 spécifient la date à laquelle le fichier a été créé ou modifié pour la dernière fois. Le format en est mm/dd/yy :  

<	21	>	<	20	>
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			m m m m		d d d d d
y y y y y y y					

 où :  
 mm vaut 1 à 12  
 dd vaut 1 à 31  
 yy vaut 0 à 119 (à partir de 1980, pour désigner les années allant de 1980 à 2099)
- Les octets 22 à 31 sont réservés à une utilisation par le système.
- L'octet 32 spécifie le numéro d'enregistrement en cours (0-127) dans le bloc en cours. On doit initialiser ce champ avant d'effectuer une opération de lecture ou d'écriture séquentielle sur la disquette : ce champ n'est pas initialisé par la fonction d'ouverture d'un fichier.
- Les 4 octets 33 à 36 spécifient le numéro d'enregistrement depuis le début du fichier (et non du bloc), en commençant par le numéro zéro. On doit initialiser ce champ avant d'effectuer une opération de lecture ou d'écriture directe sur la disquette : ce champ n'est pas initialisé par la fonction d'ouverture du fichier.  
Si la taille de l'enregistrement est moindre que 64 octets, les deux mots sont utilisés. Sinon, seuls les trois premiers octets sont utilisés.

Ouverture d'un FCB.- La zone de mémoire vive contenant un FCB doit être définie par l'utilisateur.

Celui-ci initialise un FCB *non ouvert* : le premier octet prend les valeurs 0, 1 ou 2; le nom du fichier (y compris son extension) est renseigné.

Lors de la création ou l'ouverture d'un fichier, par des fonctions de l'interruption 21h, le FCB est *ouvert* : le premier octet ne prend que les valeurs 1 ou 2 et certains des autres champs sont renseignés.

### Bloc de contrôle de fichier étendu

Un bloc de contrôle de fichier étendu est utilisé pour créer une entrée dans le répertoire de la disquette ou en rechercher une.

Il comprend un préfixe de 7 octets en plus du FCB standard, constitué de la façon suivante :

- L'octet FCB-7 est un drapeau contenant FFh pour spécifier qu'on a à faire à un FCB étendu.
- Les octets FCB-6 à FCB-2 sont réservés.
- L'octet FCB-1 spécifie l'attribut : 02h pour fichier caché, 04h pour fichier système.

Les fichiers IO.COM et MSDOS.COM sont à la fois système et cachés pour en exclure la recherche dans le répertoire. Ceci en évite l'effacement accidentel.

### 2.3.8 Les enregistrements

L'unité de lecture et d'écriture sur un fichier est l'**enregistrement** (*record* en anglais). La taille d'un enregistrement a une valeur par défaut, 128 octets, mais on peut la spécifier si nécessaire.

La fin de fichier doit donc être indiquée dans le dernier enregistrement contenant les données : il s'agit du caractère CTRL-Z en MS-DOS.

Un fichier contient un certain nombre d'enregistrements, dont la position de chacun d'eux dans le fichier est repérée par son **numéro absolu**, en numérotant consécutivement les enregistrements du fichier, en commençant par l'enregistrement de numéro zéro. Puisqu'il peut y avoir plus de 65 536 enregistrements, on a besoin de deux registres pour repérer ce numéro ; on utilise en général le couple **DX:AX**.

La position de l'enregistrement en cours, l'**enregistrement index**, est repérée soit par son numéro absolu, soit par deux champs, respectivement d'un mot et d'un octet, du FCB : le **bloc** en cours et le numéro de l'enregistrement dans ce bloc. Un bloc contient 128 enregistrements, et non 256 ; seuls les 7 bits de poids faible sont donc utilisés pour le numéro de l'enregistrement dans un bloc.

## 2.4 Appels systèmes généraux sur le système de fichiers

### 2.4.1 Opérations sur le lecteur de disquette par défaut (18h, 1Ch, 0Eh, 19h)

Fonction 24 (18h) de demande d'informations sur le lecteur de disquette en cours.- La fonction 24 (18h) de l'interruption 21h devrait permettre de demander des informations sur le lecteur de disquette en cours. En sortie, on est censé avoir les informations suivantes :

- Le registre AL indique le nombre de secteurs par unité d'allocation.
- Le registre DX indique le nombre d'unités d'allocation.
- La paire de registres DS:BX indique l'adresse du type de lecteur de disquette :
  - FFh : lecteur de disquette, double face, 8 secteurs par piste (de capacité 320 kio).
  - FEh : lecteur de disquette, simple face, 8 secteurs par piste (de capacité 160 kio).

Fonction 14 (0Eh) de sélection du lecteur de disquette par défaut.- La fonction 14 (Eh) de l'interruption 21h permet de choisir le lecteur de disquette par défaut, en le désignant dans le registre DL (avec 0 pour A et 1 pour B). Le numéro est renvoyé dans le registre AL lorsqu'il y a deux lecteurs de disquette; la valeur 2 est renvoyée sur un système comportant un seul lecteur de disquette, indiquant par là qu'il y a deux lecteurs de disquette virtuels, A et B.

L'interruption 11h du BIOS, de détermination de l'équipement, peut être utilisée pour obtenir le nombre réel de lecteurs de disquette.

Fonction 25 (19h) de détermination du lecteur de disquette par défaut.- La fonction 25 (19h) de l'interruption 21h renvoie dans AL le numéro du lecteur de disquette par défaut (en cours, avec 0 pour A et 1 pour B).

### 2.4.2 Fonction 26 (1Ah) de sélection de l'adresse de transfert (DTA)

La fonction 26 de l'interruption 21h permet de choisir l'adresse de la zone d'échange entre la disquette et la mémoire centrale, en la spécifiant par DS:DX. Le contenu transféré doit tenir dans un segment : MS-DOS ne permet pas qu'on aille au-delà.

### 2.4.3 Fonction 46 (2Eh) d'enclenchement de la vérification lors de l'écriture sur la disquette

La fonction 2Eh de l'interruption 21h permet d'enclencher la vérification lors de l'écriture sur la disquette. Lorsque la vérification est demandée, celle-ci est effectuée chaque fois qu'une opération d'écriture est effectuée sur une disquette, pour s'assurer que les données ont été proprement enregistrées. Bien que les erreurs d'enregistrement sur une disquette soient très rares, cette fonctionnalité est quand-même proposée aux utilisateurs qui veulent vérifier l'enregistrement de données critiques.

### 2.4.4 Fonction 13 (0Dh) de réinitialisation d'une disquette

La fonction 13 (0Dh) de l'interruption 21h sélectionne le lecteur de disquette A comme lecteur de disquette par défaut, initialise l'adresse de la zone de transfert à DS:80 et vide tous les tampons de fichier : en effet, les fichiers ayant changé de taille mais n'ayant pas été explicitement fermés n'étant pas proprement enregistrés sur le répertoire de la disquette, on a besoin d'appeler cette fonction avant de changer de disquette si tous les fichiers n'ont pas été correctement fermés.

### 2.4.5 Fonction 27 (1Bh) de détermination de l'adresse de la copie de la FAT en mémoire centrale

La fonction 27 (1Bh) de l'interruption 21h fait pointer DS:DX sur la copie en mémoire centrale de la FAT du lecteur de disquette en cours, place dans DX le nombre d'unités d'allocation de la disquette, dans AL le nombre de secteurs par unité d'allocation et dans CX la taille d'un secteur.



## 2.5 Opérations sur les fichiers

Voici les opérations à effectuer pour se servir d'un fichier :

- 1. Toutes les opérations sur une disquette exige un FCB proprement construit.
- 2. On doit déterminer une adresse de transfert (DTA) avant d'effectuer une lecture ou une écriture sur le fichier.
- 3. Tout fichier doit être ouvert (voire créé dans le cas d'un nouveau fichier) avant d'y effectuer une opération de lecture ou d'écriture. Un fichier sur lequel on a écrit doit être fermé afin de mettre à jour les informations du répertoire.
- 4. On peut spécifier une taille d'enregistrement en plaçant celle-ci dans le champ adéquat du FCB. Ce champ doit être rempli après l'ouverture (ou la création) du fichier mais avant toute opération de lecture ou d'écriture. La fonction d'ouverture initialise ce champ à la valeur par défaut de 128 octets.
- 5. Les nouveaux fichiers doivent être créés avant qu'on puisse écrire dessus. La fonction 16h de l'interruption 21h crée une nouvelle entrée de répertoire et ouvre le fichier.
- 6. Si la quantité de données à transférer est moindre qu'un secteur (512 octets), MS-DOS place les données dans un tampon interne.

Fonction 15 (0Fh) d'ouverture d'un fichier spécifié par un FCB.- En entrée, DS:DX pointe sur un FCB non ouvert. La fonction recherche l'entrée correspondante dans le répertoire, AL renvoyant FFh si on ne la trouve pas. Si on la trouve, AL renvoie 00h et les champs du FCB sont renseignés de la façon suivante :

- Si le champ 'code du lecteur de disquette' du FCB vaut 0 (indiquant par là le lecteur de disquette par défaut), il est changé par le lecteur de disquette en cours (1 = A, 2 = B); ceci permet de changer ultérieurement le lecteur de disquette par défaut sans interférer avec les opérations sur ce fichier.
- Le champ 'bloc en cours' (octets C-D du FCB) est initialisé à zéro.
- Le champ 'taille d'un enregistrement' (octets E-F du FCB) est initialisée par défaut à 80h (128).
- Les champs 'taille du fichier' et 'date' sont initialisés par les informations obtenues à partir du répertoire.

Il appartient à l'utilisateur de changer la taille d'un enregistrement (octets E-F du FCB) à la taille désirée si la valeur par défaut (80h = 128) n'est pas celle qu'il souhaite. Il lui appartient également d'initialiser le champ 'enregistrement aléatoire' et/ou le champ 'enregistrement en cours'. Ces actions doivent être effectuées après l'ouverture mais avant une opération sur la disquette.

Fonction 22 (16h) de création d'un fichier spécifié par un FCB.- Lors de l'appel à la fonction 16h, DS:DX doit pointer sur un FCB non ouvert. On recherche dans le répertoire de la disquette une entrée correspondant à ce FCB. Si on en trouve une, elle est réutilisée. Sinon, on recherche une entrée vide et AL renvoie FFh si on n'en trouve pas. Sinon, on initialise l'entrée avec un fichier de longueur nulle, le fichier est ouvert et AL renvoie 00h.

On peut placer l'attribut 'caché' au moment de la création en utilisant un FCB étendu contenant l'octet d'attribut approprié.

Fonction 16 (10h) de fermeture d'un fichier spécifié par un FCB.- On doit appeler la fonction 10h de fermeture d'un fichier après avoir écrit sur celui-ci, pour s'assurer des mises à jour des informations le concernant dans le répertoire. Lors de son appel, DS:DX doit pointer sur un FCB ouvert. La fonction cherche si le fichier se trouve sur le répertoire de la disquette et sa position est comparée à ce qui est indiqué dans le FCB. Si on ne trouve pas le fichier à sa position correcte dans le répertoire, on fait la supposition que la disquette a été changée et AL renvoie FFh. Sinon l'entrée de répertoire est mise à jour à partir du FCB et AL renvoie 00h.

Fonction 19 (13h) de suppression d'un fichier spécifié par un FCB.- Lors de l'appel à la fonction 13h, DS:DX doit pointer vers un FCB non ouvert. Toutes les entrées du répertoire concordant avec le nom de fichier du FCB sont effacées. Si aucune entrée concordante n'est trouvée, AL renvoie FFh, sinon AL renvoie 00h.

On parle d'entrée concordante et non de l'entrée dont le nom de fichier est exactement ce nom, car le nom peut comporter le **caractère d'ambiguïté** '?', signifiant n'importe quelle lettre.

Fonction 23 (17h) de renommage d'un fichier spécifié par un FCB.- Lors de l'appel à la fonction 17h, DS:DX doit pointer sur un FCB dont les champs 'code de lecteur de disquette' et 'nom de fichier' sont renseignés. Un second nom de fichier doit se trouver à l'emplacement d'adresse DS:DX + 11h, champ réservé du FCB. Chaque occurrence correspondant au premier nom dans une entrée du répertoire est alors changée par le second nom de fichier, en respectant le fait que deux fichiers ne peuvent pas avoir le même nom et la même extension. Si le caractère d'ambiguïté '?' apparaît dans le second nom, le caractère ayant la même position dans le nom de fichier source ne doit pas être modifié. Si aucune telle entrée n'est trouvée, AL renvoie FFh, sinon AL renvoie 00h.

Fonction 20 (14h) de lecture séquentielle dans un fichier spécifié par un FCB.- La fonction 20 de l'interruption 21h permet d'effectuer la lecture séquentielle de l'enregistrement index dans un fichier spécifié par un FCB.

Avant de l'appeler, DS:DX doit pointer sur un FCB ouvert. L'enregistrement spécifié par le bloc en cours (octets C-Dh du FCB) et l'enregistrement en cours au sein de celui-ci (octets I-Fh du FCB) est chargé à l'adresse de transfert du disque puis le numéro d'enregistrement est incrémenté. Un code d'erreur est placé dans AL :

- 00h signifie que le transfert s'est effectué avec succès.
- 01h signifie qu'il n'y avait pas de données dans l'enregistrement, c'est-à-dire qu'il ne contient que l'indicateur de fin de fichier.
- 02h signifie qu'il n'y avait pas suffisamment de place dans la zone de transfert pour lire un enregistrement et donc que le transfert n'a été que partiel.
- 03h signifie que l'enregistrement contient l'indicateur de fin de fichier et donc qu'il a été lu partiellement, suivi de zéros de remplissage.

Fonction 21 (15h) d'écriture séquentielle.- La fonction 21 (15h) de l'interruption 21h permet d'écrire un enregistrement sur un fichier spécifié par un FCB.

Lors de son appel, DS:DX doit pointer sur un FCB ouvert.

L'enregistrement spécifié par les champs 'bloc en cours' et 'numéro d'enregistrement dans ce bloc' du FCB, de taille également spécifiée par le FCB, est écrit sur la disquette depuis la zone de transfert du disque (ou, dans le cas d'enregistrements d'une taille inférieure à la taille d'un secteur, placé dans le tampon de secteur jusqu'à ce que celui-ci atteigne la taille d'un secteur). Le numéro de l'enregistrement en cours pour les opérations séquentielles est alors incrémenté.

Le code d'erreur, renvoyé dans AL, a la signification suivante :

- 00h si le transfert s'est terminé avec succès.
- 01h si la disquette est pleine.
- 02h s'il n'y avait pas suffisamment d'espace dans la zone de transfert pour écrire un enregistrement, et que le transfert a donc été interrompu.

Fonction 33 (21h) de lecture directe dans un fichier spécifié par un FCB.- La fonction 33 (21h) de l'interruption 21h permet d'effectuer la lecture directe de l'enregistrement repéré par le numéro du champ RR (Random Record) du FCB, dans un fichier spécifié par un FCB.

Avant d'appeler cette fonction, DS:DX doit pointer sur un FCB ouvert. La fonction charge au début de la zone de transfert l'enregistrement spécifié par le champ RR (octets 33 à 36 du FCB). Un code d'erreur est placé dans AL :

- 00h signifie que le transfert s'est effectué avec succès.
- 01h signifie qu'il n'y avait pas de données dans l'enregistrement, c'est-à-dire qu'on avait atteint la fin du fichier.
- 02h signifie qu'il n'y avait pas suffisamment de place dans la zone de transfert pour lire un enregistrement et donc que le transfert a été interrompu.
- 03h signifie que l'enregistrement contient l'indicateur de fin de fichier et donc qu'il a été lu partiellement, suivi de zéros.

Fonction 34 (22h) d'écriture directe dans un fichier spécifié par un FCB.- La fonction 34 (22h) de l'interruption 21h permet d'effectuer une écriture directe de l'enregistrement repéré par le champ RR (Random Record) du FCB dans un fichier spécifié par un FCB.

Avant d'appeler cette fonction, DS:DX doit pointer sur un FCB ouvert. On enregistre sur la disquette l'enregistrement se trouvant au début de la zone de transfert à l'emplacement spécifié par le champ RR (octets 33 à 36) du FCB, ou dans le tampon du secteur concerné si celui-ci n'est pas plein. Un code d'erreur est placé dans AL :

- 00h signifie que le transfert s'est effectué avec succès.
- 01h signifie que la disquette est pleine.
- 02h signifie qu'il n'y avait pas suffisamment de place dans la zone de transfert pour y placer un enregistrement et donc que le transfert a été interrompu.

Fonction 36 (24h) d'initialisation du champ RR d'un FCB.- La fonction 36 (24h) de l'interruption 21h, DS:DX pointant sur un FCB non ouvert, renseigne le champ 'enregistrement direct' (RR pour *Random Record*) à l'adresse de fichier spécifiée par les champs 'bloc en cours' et 'numéro d'enregistrement dans ce bloc'.

Fonction 39 (27h) de lecture directe de plusieurs enregistrements.- La fonction 39 (27h) de l'interruption 21h, DS:DX pointant sur un FCB non ouvert et CX contenant un nombre non nul d'enregistrements, lit ce nombre spécifié d'enregistrements depuis le numéro d'enregistrement spécifié par le champ RR du FCB et les place au début de la zone de transfert.

Un code d'erreur est renvoyé dans AL :

- 00h si tous les enregistrements ont été lus avec succès.
- 01h si on a atteint la fin du fichier avec le dernier enregistrement entièrement rempli.
- 02h si on a lu autant d'enregistrements que possible mais qu'on a rencontré le numéro de secteur FFFFh sur la disquette.
- 03h si on a atteint la fin du fichier avec le dernier enregistrement partiellement rempli.

Dans tous les cas, CX renvoie le nombre d'enregistrements lus et les champs RR, 'bloc en cours' et 'numéro d'enregistrement dans ce bloc' sont mis à jour avec le numéro d'enregistrement suivant (le premier enregistrement non lu).

Fonction 40 (28h) d'écriture directe de plusieurs enregistrements.- Le rôle de la fonction 40 (28h) de l'interruption 21h, DS:DX pointant sur un FCB non ouvert, est double :

- Si le contenu de CX est non nul, la fonction enregistre sur la disquette le nombre d'enregistrements spécifié par CX, en partant en mémoire centrale du début de la zone de transfert et en enregistrant sur la disquette à partir de l'enregistrement spécifiée par le champ RR du FCB.
- Si CX est nul, aucun enregistrement n'est écrit sur la disquette mais la taille du fichier est portée à celle spécifiée par le champ RR du FCB, qui a donc ici une signification différente de sa signification habituelle, qu'elle soit inférieure ou supérieure à la taille actuelle, et on libère ou on alloue des unités d'allocation au fichier en conséquence.

Un code d'erreur est renvoyé dans AL :

- 00h si tous les enregistrements ont été écrits avec succès.
- 01h s'il n'y avait pas suffisamment de place sur la disquette. Dans ce cas, aucun enregistrement n'a été écrit sur la disquette.

Fonction 35 (23h) de détermination de la taille d'un fichier.- La fonction 35 (23h) de l'interruption 21h, DS:DX pointant sur un FCB non ouvert, recherche sur le répertoire de la disquette

Il faut absolument que le champ 'taille d'un enregistrement' soit (correctement) renseigné avant d'appeler cette fonction ; sinon des informations erronées seront renvoyées.

Fonction 41 (29h) de création d'un FCB.- Avant d'appeler la fonction 41 (29h) de l'interruption 21h, DS:SI doit pointer sur une ligne de commande, ES:DI sur une zone de mémoire à remplir par un FCB non ouvert et AL contenir l'action à effectuer :

< ignore >  
bit : 7 6 5 4 3 2 0

- Si le bit 0 est égal 1, les séparateurs principaux devront être retirés de la ligne de commande se trouvant à l'adresse DS:SI.
- Si le bit 1 est 1, le champ 'numéro de lecteur de disquette' du FCB ne sera initialisé que si un lecteur de disquette est spécifié sur la ligne de commande.
- Si le bit 2 est 1, le champ 'nom de fichier' du FCB ne sera modifié que si la ligne de commande contient un nom de fichier.
- Si le bit 3 est 1, le champ 'extension de nom de fichier' du FCB ne sera modifié que si la ligne de commande contient une extension de nom de fichier.

Les séparateurs de nom de fichier sont les caractères suivants : '.', ',', ';', '=', '+', '/', '"', '[', ']', TAB, SPACE et tout caractère de contrôle.

La ligne de commande est analysée pour en extraire un nom de fichier spécifié sous la forme 'd:filename.ext'. Si on en trouve un, un FCB non ouvert correspondant est créé à l'emplacement ES:DI. Si aucun lecteur de disquette n'est spécifié, on doit comprendre qu'il s'agit du lecteur par défaut. Si aucune extension n'est spécifiée, on place des espaces. Si le caractère '\*' apparaît dans le nom de fichier ou dans l'extension, alors celui-ci et tous les caractères suivants du nom de fichier ou de l'extension sont des caractères d'ambiguïté '?' .

Un code d'erreur est renvoyé par **AL** :

- **01h** si l'un des caractères '?' ou '\*' apparaît dans le nom de fichier ou dans l'extension ;
- **FFh** si le numéro de lecteur de disquette n'est pas valide ;
- **00h** sinon.

**DS:SI** pointe sur le premier caractère suivant le nom de fichier et **ES:D1** sur le premier octet du FCB. Si le nom de fichier n'est pas valide, **ES:DI + 1** contient un espace.

## 2.6 Opérations sur le répertoire

Fonction 17 (11h) de recherche de la première entrée du répertoire concordant avec un nom.- La fonction 17 (11h) de l'interruption 21h cherche la première entrée du répertoire concordant avec un nom de fichier donné.

Avant d'appeler la routine, DS:DX doit pointer sur un FCB non ouvert, contenant un nom de fichier pouvant comprendre le caractère d'ambiguïté '?'.

La routine recherche dans le répertoire la première entrée dont le nom de fichier concorde avec celui passé en paramètre. Si elle n'en trouve pas, AL renvoie FFh. Sinon AL renvoie 00h et :

- Si le FCB passé en paramètre est un FCB étendu alors on place FFh comme premier octet de la zone de transfert, suivi de cinq octets nuls, de l'octet d'attribut du FCB, du numéro du lecteur de disquette utilisé (1 pour A, 2 pour B) puis des 32 octets de l'entrée de répertoire. Ainsi, la zone de transfert contient un FCB étendu non ouvert ayant les mêmes attributs que le FCB passé en paramètre.
- Si le FCB passé en paramètre est un FCB standard alors on place comme premier octet de la zone de transfert le numéro du lecteur de disquette utilisé (1 pour A, 2 pour B) puis les 32 octets de l'entrée de répertoire. Ainsi, la zone de transfert contient un FCB standard non ouvert.

Fonction 18 (12h) de recherche de l'entrée suivante du répertoire concordant avec un nom.- Si la fonction 17 de l'interruption 21h a été appelée et a trouvé une entrée concordant avec le nom de fichier (pouvant contenir des caractères d'ambiguïté '?') spécifié par le FCB, on peut appeler la fonction 18 (12h) de l'interruption 21h pour trouver l'entrée suivante du répertoire concordant avec ce nom ambigu.

Avant d'appeler la fonction, DS:DX doit pointer sur un FCB non ouvert, contenant un nom de fichier comprenant le caractère d'ambiguïté '?'.

La routine recherche dans le répertoire l'entrée suivante dont le nom de fichier concorde avec celui passé en paramètre. Si elle n'en trouve pas, AL renvoie FFh. Sinon AL renvoie 00h et :

- Si le FCB passé en paramètre est un FCB étendu alors on place FFh comme premier octet de la zone de transfert, suivi de cinq octets nuls, de l'octet d'attribut du FCB, du numéro du lecteur de disquette utilisé (1 pour A, 2 pour B) puis des 32 octets de l'entrée de répertoire. Ainsi, la zone de transfert contient un FCB étendu non ouvert ayant les mêmes attributs que le FCB passé en paramètre.
- Si le FCB passé en paramètre est un FCB standard alors on place comme premier octet de la zone de transfert le numéro du lecteur de disquette utilisé (1 pour A, 2 pour B) puis les 32 octets de l'entrée de répertoire. Ainsi, la zone de transfert contient un FCB standard non ouvert.

La zone réservée du FCB conserve les informations nécessaires pour continuer la recherche, donc aucune opération ne doit être effectuée sur la disquette avec ce FCB entre un appel précédent à la fonction 11h ou 12h et celui-ci.

## 2.7 Gestion des programmes

### 2.7.1 PSP d'un programme

Toute instance de programme en mémoire centrale est précédée d'un en-tête, ou **PSP** (*Program Segment Prefix*) de 256 octets :

- 00h-01h : appel à l'interruption INT 20h de terminaison d'un programme.
- 02h-03h : adresse du segment de la zone de mémoire centrale contenant le premier octet du programme.
- 04h : octet réservé auquel on n'attribue pas de valeur particulière.
- 05h-09h : appel long au point d'entrée dans le programme.
- 0Ah-0Dh : adresse de terminaison du programme précédent (soit ancien INT 22h).
- 0Eh-11h : adresse de sortie sur CTRL-C du programme précédent (soit ancien INT 23h).
- 12h-15h : adresse de sortie après une erreur critique du programme précédent (soit ancien INT 24h).
- 16h-17h : segment du PSP père (en général COMMAND.COM).
- 18h-2Bh : table **JBT** (pour *Job File Table*) de 20 octets.
- 2Ch-2Dh : adresse du segment d'environnement.
- 2Eh-31h : adresse SS:SP du dernier appel à INT 21h.
- 32h-33h : taille de la JFT.
- 34h-37h : pointeur sur la JFT.
- 38h-3Bh : pointeur sur le PSP précédent (utilisé uniquement par SHARE à partir de MS-DOS 3.3).
- 3Ch-3Fh : 4 octets réservés.
- 40h-41h : version du MS-DOS de retour (à partir de MS-DOS 4, altérable par SETVER à partir de MS-DOS 5).
- 42h-4Fh : 14 octets réservés.
- 50h-52h : appel lointain à MS-DOS (contient toujours INT 21h + RETF).
- 53h-54h : 2 octets réservés.
- 55h-5Bh : 7 octets réservés (pouvant être utilisés pour que le premier FCB soit un FCB étendu).
- 5Ch-6Bh : 16 octets constituant le premier FCB standard non ouvert.
- 6Ch-7Fh : 20 octets constituant le deuxième FCB non ouvert (détruit si le FCB 1 est ouvert).
- 80h : nombre d'octets de la ligne de commande.
- 81h-FFh : 127 octets contenant la ligne de commande (se terminant par 0Dh).

Le PSP est une réminiscence de la *Zero Page* du système d'exploitation CP/M.

### 2.7.2 Fonction 00h et interruption 20h de terminaison d'un programme

La fonction 00h de l'interruption 21h effectue exactement la même chose que l'interruption 20h. Les adresses de fin et de sortie sur CTRL-BREAK sont restaurées aux valeurs qu'elles avaient lors de l'entrée dans le programme à la fin du programme, à partir des valeurs sauvegardées dans le PSP. Tous les tampons sont vidés, mais les fichiers dont la longueur a changé et non fermés ne seront pas enregistrés correctement dans le répertoire. Le contrôle est transféré à l'adresse de terminaison.

Il est de la responsabilité du programme de s'assurer que le registre CS contient l'adresse de segment de bloc de contrôle de son PSP avant d'appeler cette fonction.

### 2.7.3 Fonction 37 (25h) d'initialisation d'un vecteur d'interruption

Les vecteurs des interruptions ont été initialisés lors de l'initialisation du système d'exploitation mais certaines applications désirent changer les routines de service de celles-ci.

La fonction 37 (25h) de l'interruption 21h permet de placer l'adresse contenue dans DS:DX comme vecteur de l'interruption dont le numéro est contenu dans le registre AL.

### 2.7.4 Fonction 38 (26h) de création d'un nouveau PSP

La fonction 38 (26h) de l'interruption 21h crée un nouveau PSP.

Avant d'appeler cette routine, DX doit contenir le numéro de segment pour mettre en place un nouveau segment de programme. Les 256 premiers octets du programme en cours sont copiés dans le segment du nouveau programme. L'information sur la taille en mémoire se trouvant à l'emplacement 6 est mis à jour dans le nouveau segment, les adresses de terminaison et de la routine de service de CTRL-BREAK (obtenues à partir de la table des vecteurs d'interruption pour les interruptions 22h et 23h) sont placées dans le segment du nouveau programme au décalage 0Ah. Elles seront restaurées depuis cette zone lorsque le programme se terminera.



## 2.8 Bibliographie

[IBM-82] **Disk Operating System, version 1.10**, IBM 6172220, Personal Computer, Computer Language Series, second edition, May 1982, xii + 8 + 30 + 67 + 34 + 28 + 50 + 28 + 13 + 19 + 20 + 11 + 3 + 6 + 13 p.

[L'appendice D donne la description officielle des interruptions de DOS 1.0.]