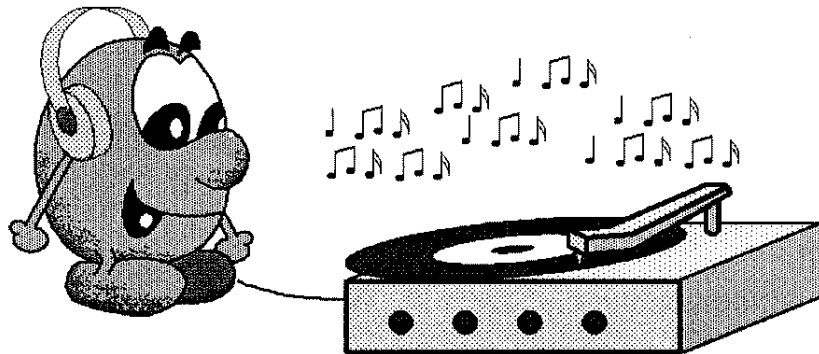


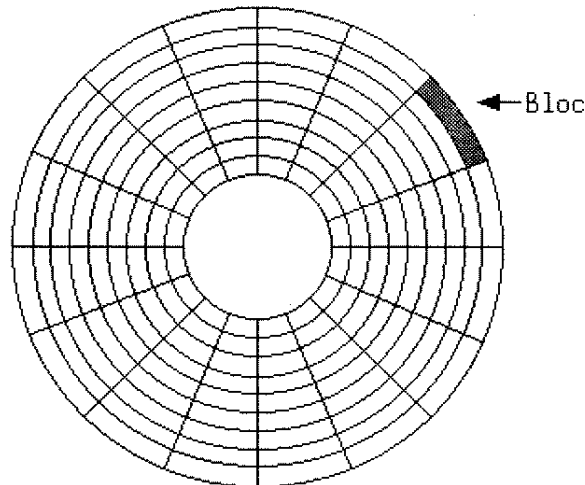
Smaky info

Les disquettes par P.-O. Vallat



Organisation d'une disquette

Une disquette est organisée en blocs: ce sont des unités de 256 octets.



Au début du disque, on trouve 16 blocs (H'10) de démarrage, générés par la commande GENFBOOT. Si ces 16 blocs sont vides, il n'est pas possible de démarrer avec la disquette; message: "Programme de démarrage inexistant".

Pour un disque dur, le nombre de blocs de démarrage est porté à 32 (H'20).

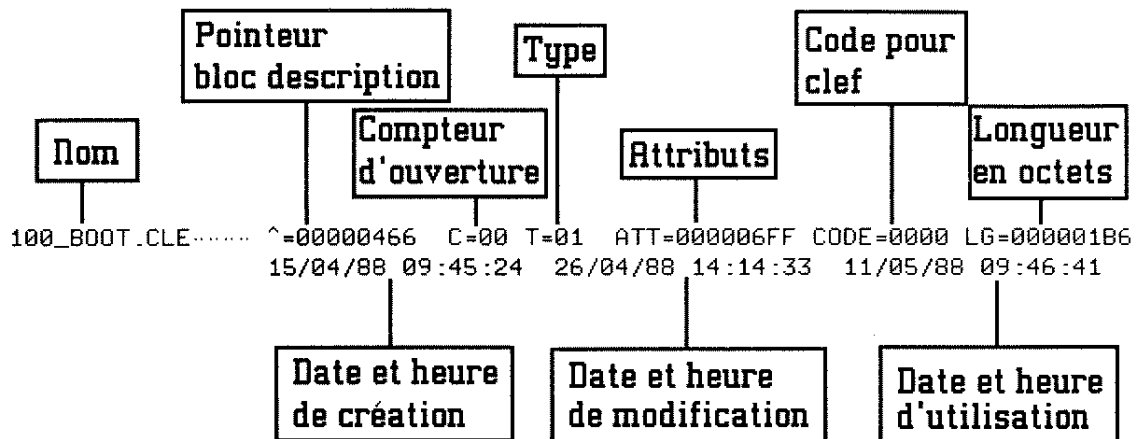
Le bloc 17 (H'11), respectivement 33 (H'21) pour un disque dur, est le bloc de description du fichier SYS_LIST (cf ci-dessous).

Le fichier SYS_LIST débute au bloc 18 (H'12) (34 (H'22) pour un disque dur). Il s'agit des blocs de liste.

SYS_LIST (ensemble des blocs de liste)

Le fichier SYS_LIST contient le nom, les attributs, la longueur, les dates de création, de dernières modification et utilisation de tous les fichiers présents sur la disquette. Il contient en plus le pointeur (adresse) du bloc de description de chaque fichier. Un bloc de liste peut contenir les informations de 4 fichiers.

Exemple:



Compteur d'ouverture

Indique le nombre de programmes utilisant le fichier et doit être à zéro. Il peut être remis à 0 en utilisant l'ordre *CLEAR nom_du_fichier* depuis le CLE.

Attributs

Déterminent si un fichier peut être détruit, lu, modifié... On peut changer les attributs d'un fichier depuis *START.CODE* ou *FILER.CODE* en utilisant l'ordre *CHATR*. Il est préférable de ne pas modifier cette valeur depuis *EDISK*, en cas de nécessité copier le nombre donné par un fichier de même type.

Code pour la clef

Il ne faut pas modifier ce nombre. Un fichier non-protégé contient toujours 0. C'est l'opération verrouille ou déverrouille du *START* qui modifie cette valeur

Longueur en octets

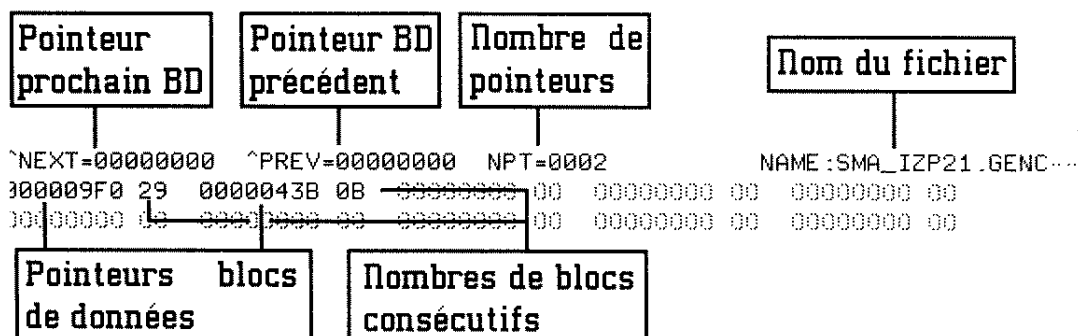
Il s'agit du nombre exact d'octets contenus dans le fichier. On peut connaître ce nombre en utilisant l'option "Modifie" dans *START.CODE*.

⚠ Les nombres rendus dans *EDISK* sont en hexadécimal!

Bloc de description

Le fichier *SYS_LIST* contient le pointeur du bloc de description. Pour économiser de la place sur une disquette, un fichier ne sera pas obligatoirement mémorisé de façon séquentielle, c'est-à-dire tous les octets à la suite les uns des autres. Le système d'exploitation essaie toujours de combler les trous provoqués par l'utilisation d'une disquette, en particulier par les destructions. Le fichier à sauver sera donc fragmenté en plusieurs portions! Le bloc de description contient les adresses et les longueurs des différents morceaux.

Exemple:



Pointeur prochain bloc de description (BD) et précédent BD

Si le fichier doit être fragmenté en plusieurs morceaux, il peut arriver qu'un seul BD ne suffise pas. Le dernier pointeur, s'il n'est pas nul, contient le pointeur du prochain BD. On peut donc passer de l'un à l'autre BD en utilisant les valeurs contenues dans NEXT et PREV.

NPT (nombre de pointeurs)

Ce nombre **doit correspondre** au nombre affiché de pointeurs aux blocs de données.

NAME

Ce nom **doit être le même que celui figurant dans SYS_LIST**. Si ce n'est pas le cas, le fichier apparaît normalement à l'affichage dans START, mais START affiche "Fichier inexistant" si l'on tente de l'exécuter! Il n'est plus possible de le détruire depuis START, ni depuis FILER.

Pointeur de bloc

Donne l'adresse du début des blocs de données.

Nombre de blocs consécutifs

Un "morceau de fichier" contient en général plusieurs blocs suivis! Ce nombre indique combien de blocs contigus appartiennent au même fichier.

Blocs de données (DATA)

Contenu du fichier lui-même.

No du bloc sur la disquette	Nom du fichier, doit correspondre à celui de SYS_LIST!	No du bloc dans le fichier
------------------------------------	---	-----------------------------------

Bloc de data numéro 000000C4 (000000B4) fichier: EDISK03.IMAGE gpos: 00000000

```

81040102 026800C8 000014F9 00000000 00000000 00000000 00000000 00000000
00000000 1B001500 55550005 50000000 01000000 00001500 FF95FC00 0F0CFFFC
010CFFF8 1222 180C0000 2A000C0C 18220018 0C000C2A
000C0180 0B00 83FC0300 7004001C
000C0C00 0C01 FF0C0183 FE030070
00141822 0018 00018003 0007FC04
0204001C 00182200 18FF8088 06000607 000C0000 3F10E001 80018003 0007FC04
FF8C0181 86030030 0400050C 00182200 18618002 00110607 000C0000 18600200

```

Contenu du fichier en hexadécimal

```

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

```

Contenu du fichier en ASCII

```

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

```

Systemes de sécurité des disquettes

Checksum

Lors de l'écriture d'un bloc sur la disquette un nombre appelé CHECKSUM est écrit pour s'assurer de la validité des données. Plusieurs systèmes existent. Lors de la lecture des données, la checksum est reconstituée et doit correspondre à celle figurant sur la disquette. Si ce n'est pas le cas, les données lues ne sont pas valables ("Erreur de lecture").

Exemple de checksum pas utilisée en informatique: la preuve par 9

On doit inscrire les nombres 117, 178 et 222 sur une disquette. Les trois nombres sont inscrits et la somme est calculée: 517. La preuve par 9 donne: 4. Le nombre 4 est inscrit sur la disquette. Lors de la lecture, les nombres 117, 178 et 222 sont lus.

La somme est faite, ainsi que la preuve par 9. Le programme obtient 4 et la checksum est lue sur la disquette. Si les deux nombres sont identiques, la lecture est considérée comme correcte sinon elle n'est pas acceptée. C'est le gestionnaire de bas niveau qui se charge de ce travail - BIOS Basic Input Output System.

Numéro de contrôle pour les blocs de liste et les BD

Les BL et BD sont vitaux pour la gestion d'une disquette. Ils sont protégés par un numéro de contrôle qui doit être le même pour tous les BL et les BD. Cette gestion est faite par le FOS - File Operating System. Si le numéro de contrôle d'un BL ou d'un BD n'est pas celui de la disquette, cela signifie qu'il est étranger à la disquette, d'où risque de mélange!

Cela se produit lorsqu'une disquette A est sortie du lecteur sans désactivation et remplacée par une disquette B. Si l'utilisateur sauve un fichier sur cette disquette B, c'est le numéro de contrôle de la disquette A qui sera inscrit, car pour le FOS c'est toujours la disquette A qui est dans le lecteur puisqu'il n'y a pas eu de désactivation!. *Bonjour la pagaille!*

SYS_FREE Lors du sauvetage d'un fichier, la recherche des blocs libres serait trop longue s'il fallait les retrouver parmi tous les BD déjà existants. SYS_FREE contient la liste de tous les blocs occupés sous forme d'un bit-map. Lors de l'activation, ce fichier est chargé en mémoire vive et continuellement mis à jour par le FOS. Lors de la désactivation, ce fichier est à nouveau inscrit sur la disquette. Si l'on oublie de désactiver une disquette le programme SMA_RECUP.CODE reconstituera ce bit-map.


Tentatives de récupération d'une disquette

CLE/FILER

Erreur	"Fichier utilisé"	Clear NOM_DU_FICHER (CLE ou FILER) referme le fichier.
	Clear * (FILER)	Referme tous les fichiers ouverts sur la disquette.
Erreur	"Lecture impossible"	Delete NOM_DU_FICHER (CLE ou FILER) détruit le fichier. L'attribut de destruction doit être présent.

SMA_RECUP

Disquette pas désactivée S'enclenche automatiquement si la disquette n'a pas été désactivée lors de la dernière utilisation.


 En tapant "SMA_RECUP/D/V" dans le CLE, on rend le programme plus bavard!

Erreur grave sur le disque SMA_RECUP signale les erreurs graves qui ne peuvent être réparées que par EDISK. Il faut noter les numéros des blocs signalés comme étant défectueux.

TFLO Permet de tester physiquement une disquette en cas d'erreur de lecture.

Faire tester la disquette par TFLO, puis demander l'option "précisions" et imprimer la liste des erreurs. On y trouve les numéros des blocs défectueux.

Si l'on veut utiliser EDISK pour restaurer la disquette, il ne faut pas oublier d'ajouter 10 au numéro hexadécimal de chaque bloc donné par TFLO.

 Avant de lancer le programme EDISK, il est conseillé de faire une copie de la disquette défectueuse! Lorsqu'une disquette ne peut plus être activée, il n'est plus possible de la copier depuis START.

FCOPY Si l'erreur interdisant l'activation de la disquette n'est pas due à une erreur de lecture, il est possible d'utiliser FCOPY.CODE pour copier la disquette défectueuse sur une autre disquette.

COPY_DET.CODE Le programme COPY_DET permet de copier une disquette sur une autre disquette ou dans un fichier .DI. Il faut impérativement utiliser un poste avec deux lecteurs de disquettes ou un poste avec disque dur et un lecteur de disquettes.

Après le lancement du programme, des petits points s'affichent. Chaque petit point représente une série de 10 blocs lus correctement. Lorsque le programme rencontre une difficulté de lecture, il lit les blocs les uns après les autres. Si un signe "+" est affiché, cela signifie qu'un bloc a été lu correctement. Si un signe "-" s'affiche en vidéo-inverse, la lecture n'a pas été faite correctement (malgré 10 essais). Le contenu du bloc copié est donc sujet à caution!

A la fin du programme, la liste des blocs défectueux est affichée à l'écran. Il faut absolument la copier ou la faire imprimer par le programme. En effet les blocs défectueux n'apparaissent plus sur la copie. Pour voir si leur contenu est valable, il faut donc les visualiser dans EDISK.

EDISK sera décrit dans le prochain SMAKYinfo.