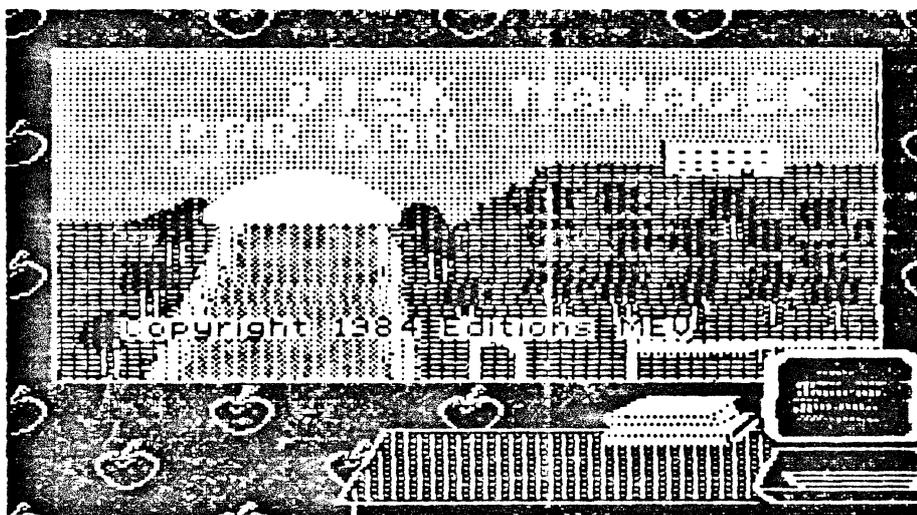


Le Disk Manager :

une nouvelle approche pour la gestion du disque ...

Le Disk Manager est un langage permettant de développer la capacité du Disk Operating System (DOS) et d'élaborer aisément des programmes utilitaires.

La présente disquette contient, outre le Disk Manager et ses programmes propres, quatre programmes d'applications importants, se rapprochant des programmes les plus vendus sur le marché. Ils ne constituent qu'un banc d'essai des possibilités du Disk Manager...



Avertissement

Les instructions du Disk Manager vous permettent de modifier l'organisation de la disquette. Il est donc fortement conseillé de faire l'apprentissage du Disk Manager sur une copie de disquette, afin de ne pas risquer de détruire ou dénaturer une disquette par inadvertance.

Editions MEV - 49 rue Lamartine - 78000 Versailles

Copyright Dan Steerey et Editions MEV

SOMMAIRE

	Page
I - Introduction	
Le DOS en kit	3
II - Analyse d'une disquette	4
Le catalogue	5
Les TSL	6
La carte d'occupation	7
Les différents types de fichiers	8
III - Description générale	10
Syntaxe des commandes	11
Les buffers et HIMEM	12
IV - Commandes de lecture/écriture	13
LIRE	14
ECRIRE	14
LIRIND	14
AFFECTER	15
V - Modification d'un secteur	16
EDITER	16
VI - Commandes de carte et de nom	22
1- Allocation et libération de secteurs	23
LIRCARTE	23
ECRCARTE	23
CARTE	24
INSCARTE	24
SUPCARTE	24
ESPACE	24
2- Commandes de noms	24
LIRNOM	24
CHRGNOM	25
ECRNOM	25
SAUVNOM	25
LIRADR	26

VII - Premières applications	27
1- Applications simples	27
Le catalogue dans un tableau	27
Réécriture du catalogue	27
LOCK/UNLOCK en série	28
DOS absent	28
Classement du catalogue	29
La carte d'occupation sur l'écran	30
2- Applications plus élaborées	30
FIX, correction de la longueur d'un programme	30
Le SCANNER	31
47 secteurs en plus	32
UNDELETE, restauration de programme	32
Reconstruction de la carte d'occupation	32
Copie piste à piste	33
Copie fichier par fichier	34
Emploi des commandes du Disk Manager	35
VIII - Programmes utilitaires généraux	36
1- ULTRA-COPIE	36
2- UTILI-DISQUE	36
IX - EDICAT	39
X - MULTI DISQUE	40
 Annexes	 41
1) Résoudre et prévenir les problèmes	43
2) Les adresses du DOS 3.3 et leur utilisation	43
3) Carte de la mémoire	48

I - Introduction

Le DOS en kit

Le Disk Manager est le premier programme qui décompose les fonctions essentielles de l'accès au disque en commandes. Des commandes simples d'emploi, qui permettront au programmeur amateur de réaliser sans apprentissage particulier tous les utilitaires disque dont la publicité nous décrit les possibilités alléchantes.

Il se présente sous la forme d'un jeu d'instructions qui, effectuant les opérations de base du Disk Operating System (DOS), permettraient de reconstituer toutes ses commandes telles "LOAD", "OPEN", "READ" etc.. d'où le terme de "kit", et dont l'intérêt essentiel sera de rendre possible la réalisation en Basic d'utilitaires élaborés, sans exiger pour autant une connaissance approfondie du DOS 3.3 ni le recours à l'assembleur.

Il sera également utile au programmeur expérimenté, en assembleur notamment, car il lui permettra de faire l'essai d'algorithmes en Basic, qui seraient impossibles sans les commandes du Disk Manager et que l'on pourra, une fois mis au point, traduire en assembleur. Cette conversion semble particulièrement indiquée pour les programmes effectuant des opérations simples mais en grande série alors que les programmes plus complexes requièrent au contraire le langage Basic comme langage final, associé au Disk Manager.

Ce manuel comprend une description détaillée de la structure d'une disquette, des "track/sector lists" et du DIRECTORY qui comprend le catalogue et la "VTOC", c'est-à-dire la carte d'occupation de la disquette.

Ces notions cependant ne sont pas nécessaires à l'emploi du kit; elles ont pour objectif d'informer les amateurs désireux de comprendre comment ce petit disque de plastique souple mémorise les données qu'on lui transmet et les restitue à la demande.

Le programme DOM (Disk Organizer Manager), contenant en 1.75 K toutes les commandes d'accès disque peut être employé pour réaliser une organisation personnalisée de la disquette, optimisée en fonction d'un traitement donné, par exemple un accès ultra-rapide aux fichiers.

La fonction "EDITER" opère comme un microscope placé sur le disque, que l'on oriente avec deux touches pour visualiser, secteur par secteur, agrandi des milliers de fois, le contenu de la disquette en code ASCII ou hexadécimal. Cette fonction EDITER possède bien entendu des commandes permettant de modifier le contenu de ces secteurs en mémoire et de les réécrire. C'est en fait un utilitaire d'accès disque complet, qui présente l'avantage d'être aussi commandé par programme.

Il va sans dire que ces dernières opérations supposent quant à une bonne connaissance de la structure d'une disquette; de nombreuses indications pour réaliser des opérations qui nécessitent habituellement des programmes spécialisés pourront d'ailleurs être trouvées dans ce manuel.

Les programmes en Applesoft qui accompagnent le Disk Manager des utilitaires de gestion du disque d'un emploi fréquent; ils sont parfois originaux et représentent des opérations quelque peu délicates. Mais je dois dire que j'ai pu les réaliser en un temps assez court à l'aide des instructions fournies par le Disk Manager dont ils ont constitué le banc d'essai.

Le Disk Manager est donc à la fois un système aux possibilités diverses et, en combinaison avec les instructions de l'Applesoft langage. Avec une spécialisation particulière sur l'accès disque est dans la lignée des logiciels tels que Visicalc : utile au grand public le plus large, sans toutefois être limité pour l'expert. Ces logiciels programmables constitueront ensemble, avec leur nombreux développements futurs, la réponse à la question "Quel sera le langage de demain ?" Les langages classiques, souvent lourds à mettre en œuvre, apprendront voyant leur usage restreint à l'emploi des gros ordinateurs.

II - Analyse d'une disquette

Le vocabulaire relatif au DOS de l'Apple II, le DOS II, est impressionnant par une série de termes techniques aussi nombreux qu'ils semblent inaccessibles : VTOC, TSL, DIRECTORY, etc... L'organisation de l'occupation d'une disquette est cependant compréhensible par tout utilisateur.

La disquette est divisée en 560 secteurs placés de façon circulaire sur 35 pistes, à raison donc de 16 secteurs par piste.

Le DOS est normalement présent sur chaque disquette, dans les pistes 0, 1 et 2, à partir desquelles il est chargé en mémoire lors du "boot". Le catalogue est placé sur la piste 17, au centre de la disquette, et occupe les secteurs 15 à 1, en ordre décroissant. L'état de chaque secteur, libre ou occupé, est indiqué sur la piste d'occupation, dite "Volume Table Of Contents" ou VTOC, placée sur le secteur 0 de cette même piste 17. L'ensemble formé par le catalogue et le VTOC est ce qu'on appelle le DIRECTORY.

Tout programme ou fichier stocké sur la disquette possède la liste de ses secteurs propres, la "TRACK SECTOR LIST" ou TSL, à laquelle on fera souvent référence dans ce manuel. C'est la TSL d'un fichier qui indique au DOS où le trouver sur la disquette quand ce programme ou fichier est appelé en mémoire.

On constate sur le catalogue, en stockant un nouveau programme, que la taille peut être contenue dans un seul secteur, qu'il en soit quand même deux. Le second est réservé à sa TSL, qui peut avoir une référence à 122 secteurs avant d'être dédoublée par le DOS.

Chaque type de programme est mémorisé sur la disquette avec certaines références : les programmes en Applesoft et Integer sont mémorisés avec leur longueur, les programmes binaires avec leur adresse de départ et leur longueur, et les fichiers sans rien de particulier, la fin de fichier étant détectée par la première occurrence du code "0" sur un secteur.

Le premier secteur d'un programme binaire présente un format de ce type :

```
0008002001020406 etc...
! ! !
! ! début du programme
! !longueur ($2000)
!adresse de départ ($800)
```

Le catalogue

15 secteurs sont réservés au catalogue, ce qui à raison de 7 noms par secteur permet le stockage de 105 programmes; si cette possibilité était entièrement utilisée, cela supposerait une longueur moyenne de moins de 5 secteurs par programme, soit moins de 4 sans la TSL : on voit donc que la capacité du catalogue est largement suffisante et même qu'elle est toujours sous-utilisée. Pourquoi ne pas en récupérer une partie pour stocker les programmes ? Ceci fera l'objet d'une application décrite dans les pages suivantes.

Chaque nom dans le catalogue est accompagné de quatre renseignements : le numéro de piste et de secteur de la TSL, le type du programme et sa longueur. Les codes identifiant chaque type de fichier sont les suivants :

```
0 : fichier
1 : Integer
2 : Applesoft
4 : binaire
8 : fichier source
16 : programme relogeable
```

Un secteur de catalogue se présente de la façon suivante. Position :

```
0 : FF si occupé, 0 autrement
1 : no piste du secteur suivant
2 : no secteur de la suite du catalogue
3 : no piste du secteur lui-même
4 : no secteur correspondant
```

Les positions 5 à 10 sont inutilisées et le reste du secteur se divise en 7 zones de 35 octets identiques, dont le contenu est le suivant, des positions 0 à 34.

```
0 : no piste de la TSL
1 : no secteur correspondant
2 : type, (+ 128 si LOCKE)
```

3 à 32 : nom du programme, complété par des espaces
 33 : longueur, partie basse
 34 : longueur, partie haute possible mais non utilisée par le 3 :

Le nom du programme est mémorisé avec le bit le plus significatif 1 alors que le format interne de l'Applesoft suppose que toutes les variables alphanumériques soient stockées en mémoire avec ce bit ce que l'Applesoft corrige lors de l'affichage. Cette différence doit être notée pour la suite.

Premier secteur du catalogue, représenté selon le format de commande EDITER, mais en mélangeant l'affichage caractères et hexadécimaux:

```
PST:$11 SEC:$0F DRV:1 BUF:1 DRV:oui
PAGE:$9600-38400 F:n I:n C:n LOCK
```

```
000102030405060708090A0B0C0D0E0F
-----
00!FF110E110F0000000000000120082 H E
01! L L O
02! 0500130000 D
03! A T A
04! 9A00130884
05! S U B
06! 0400
07!
```

Le programme HELLO est en Applesoft, verrouillé; son stockage est sur la piste 12, secteur 0, où se situe donc sa TSL; sa longueur indiquée après le nom, est de 5 secteurs.

Le fichier DATA est stocké à partir du secteur 0 de la piste et occupe 154 (\$9A) secteurs.

Le programme binaire SUB est verrouillé, et a sa TSL sur le secteur 8 de la même piste 13.

Chaque fois que la commande CATALOG est utilisée par l'opérateur DOS charge auparavant la carte d'occupation dans un buffer; pour les renseignements qu'elle fournit, celle-ci indique l'adresse du premier secteur du catalogue, ce qui signifie que l'on peut reloger ce catalogue en indiquant ici sur quelle piste et secteur il débutera ! Ceci est très flexible du DOS, que l'on vérifiera souvent par ailleurs, mais la voie à de larges applications mais exige en contrepartie que l'on consulte des listes de paramètres telles que celle contenue dans la carte d'occupation.

Les TSL

Les positions 0, puis 3 à 11, sont inoccupées sur une TSL.

En position 1, on trouve le numéro de piste de la TSL suivant; en position 2 son numéro de secteur. Ces positions contiennent 0 si la TSL est la seule ou la dernière.

En position 12 se trouve le numéro de piste du premier secteur du programme, en 13 son numéro de secteur; les coordonnées des secteurs suivants occupent tous les octets qui suivent : il y a dorénavant

façon définitive et sans risques, l'application UNDELETE du chapitre VII vérifie qu'aucun secteur du programme n'a été réutilisé depuis le DELETE; si tout va bien, tous les secteurs du programme sont alors réintégrés sur la carte.

III - Description générale

La division du programme en deux parties permet en premier lieu d'avoir en mémoire un système d'accès disque économe en espace : le DOM contient toutes les commandes disque en 1,75 K - Ensuite, complémentaire mais chargé séparément, un éditeur de secteurs, le DE (Disk Editor) ne contient que la commande EDITER et occupe 2,5 K.

Le Disk Manager possède un interpréteur propre, complémentaire à celui de l'Applesoft et très rapide. La conception du système réalise le meilleur compromis espace/vitesse en considération de chaque commande :

Les commandes LIRE, ECRIRE, CARTE, INSCARTE, SUPCARTE et LIRIND sont des commandes essentiellement rapides, car elles doivent s'exécuter dans des boucles répétitives; toutes les autres commandes ont été optimisées en vue de l'économie d'espace, pour que le DOM occupe moins de 2 K octets.

L'éditeur de secteurs, quant à lui, recherche avant tout la vitesse d'affichage, l'économie de place comptant moins dans son cadre d'utilisation. Il peut s'appeler avec la commande EDITER incluse dans l'interpréteur de DOM ou, si DE est chargé seul, par CALL 10 en Basic, ou *AG à partir du moniteur.

Les deux parties sont chargées par BRUN DOM d'abord, puis BRUN DE : elles sont totalement indépendantes, mais co-résidentes en mémoire, et possèdent une zone de paramètres commune.

DE occupe les positions \$8500 à \$8EFF, DOM les positions \$8F00 à \$95FF, et le sommet de la mémoire est abaissé automatiquement à l'adresse de départ quand on charge l'un d'eux.

La zone de paramètres qui interface les deux parties occupe les positions 24 à 31; elle peut être remplie avant un appel direct par CALL 10, ou par un programme Basic, sous réserve de respecter l'allocation suivante :

24 ou \$18 : numéro de piste
25 ou \$19 : numéro de secteur
26 ou \$1A : numéro de drive connecté
27 ou \$1B : premier buffer, partie basse
28 ou \$1C : partie haute
29 ou \$1D : numéro de buffer affiché
30 ou \$1E : second buffer, partie basse
31 ou \$1F : partie haute

52 : nombre de piste par disquette (35 ou 36)
 53 : nombre de secteurs par piste (13 ou 16)
 54 : nombre d'octets par secteur, partie basse
 55 : partie haute de ce nombre
 56-209 : tableau d'occupation pour 35 pistes
 210-11 : inutilisés
 212-13 : allocation d'une 36ème piste
 214-56 : inutilisés

L'allocation de chaque secteur de la disquette est indiquée sur 2 octets, séparés de l'allocation suivante par 2 octets libres. Le premier donne l'allocation des secteurs 15 à 8, le second celle des secteurs 7 à 0. Cela donne le schéma suivant en hexadécimal (F=15, E=14, ...)

1	2	3	4
-----	-----	-----	-----
FEDCBA98	76543210	00000000	00000000

On voit que les huit bits de l'octet 1 et de l'octet 2 correspondent chacun à un secteur de la piste concernée; ils sont positionnés à 0 si le secteur est utilisé, à 1 s'il est libre.

L'utilisateur du Disk Manager n'aura pas à se préoccuper de la manipulation complexe de ces octets, car il disposera des commandes suivantes du Disk Manager, qui seront décrites plus en détail ultérieurement :

- INSCARTE, suivi d'un numéro de piste et de secteur, met la carte à jour en indiquant le secteur spécifié comme occupé
- SUPCARTE identifie, de la même façon, un secteur comme libéré
- CARTE permet de savoir si un secteur est libre ou non

Les différents types de fichiers

Les programmes Applesoft

Contrairement au Basic Integer, qui place les programmes en sommet de mémoire, l'Applesoft les place au début de la mémoire centrale, au-dessus de la mémoire écran, à partir de l'adresse \$800 (2048) ou de l'adresse pointée en 103-104. Le premier code du programme est un 0 et chaque ligne est terminée par le code 0. Les pointeurs 175-176 contiennent l'adresse de fin du programme où sont placés trois zéros. Les variables sont situées au-dessus de celui-ci, entre les adresses pointées en 105-106 et 115-116. La sauvegarde d'un programme Applesoft enregistre sur la disquette la partie de mémoire contenue entre les adresses pointées en 103-104 et 175-176, sans regard au contenu. Le chargement se fait également à l'adresse pointée en 103-104.

Les programmes binaires

Le chargement d'un programme binaire est fonction uniquement des données qui sont stockées avec lui; si la commande ne spécifie pas une adresse de chargement, il se fait à partir de l'adresse de départ indiquée, la longueur enregistrée servant à vérifier que la taille

mémoire est suffisante.

Les fichiers séquentiels

Aucune référence n'accompagne un fichier séquentiel, les enregistrements étant stockés à la suite l'un de l'autre, délimités par le code \$8D, ou RETURN. Le dernier enregistrement est suivi du code 0.

Fichiers à accès direct

La lecture et l'écriture d'un enregistrement s'opèrent par calcul d'adresses, effectué par le DOS, en fonction de la longueur des enregistrements. Un enregistrement peut dépasser la longueur spécifiée; il sera alors lu et écrit correctement, mais au détriment de l'enregistrement qui le suit. Le délimiteur \$8D est également présent et doit être pris en compte dans la définition des longueurs d'enregistrement.

Les types S et R, et nouveaux A et B

Les deux premiers types de fichiers sont reconnus par la commande CATALOG, mais n'ont pas d'autre caractéristique pour le DOS, si ce n'est qu'on ne peut y accéder par les ordres de chargement habituels.

Les programmes de type R sont reconnus par le DOS ToolKit comme modules relogeables et sont réassemblés à l'adresse donnée.

Pour créer des fichiers d'un nouveau type, on peut, en utilisant les trois autres codes de types par la commande ECRNOM du Disk Manager, écrire une procédure d'implantation du fichier. On peut aussi utiliser la procédure de stockage du DOS 3.3 en sauvegardant un programme, binaire par exemple, puis en rectifiant son type avec la commande ECRNOM (voir chapitre VI) et en utilisant sa TSL comme guide de stockage. On peut consulter, pour se renseigner sur le sujet, l'étude de la TSL, et l'application FIX qui décrit la lecture d'une TSL.

LOCK, UNLOCK, DELETE

Le verrouillage d'un programme est spécifié par l'octet de type de ce programme, en position 2 sur le catalogue, dont la valeur est de 128+x quand un programme est verrouillé, x seulement autrement, x étant le numéro de type.

Le DOS 3.3 utilise, pour supprimer un programme du catalogue, une procédure simplifiée qui consiste à remplacer le numéro de piste de la TSL par 255, et à placer ce numéro de piste en dernière position du nom du programme, donc en position 30 du nom et 33 de la zone de description du programme. En outre, tous les secteurs occupés par ce programme sont déclarés libres sur la carte d'occupation de la disquette.

Restaurer le programme peut donc se faire simplement en remettant le numéro de piste de la TSL à sa place. Cependant, pour le restaurer de

La partie haute, rappelons-le, est la valeur d'une adresse divisée par 256, la partie basse étant le reste de la division. Ce reste sera toujours 0 en ce qui concerne les adresses de buffers pour la fonction EDITER.

Le numéro de buffer est 1 ou 2 selon que l'on souhaite démarrer en affichant le premier ou le second; il pourra ensuite être modifié par les commandes internes de EDITER.

Les numéros de piste et secteur ne sont POKés directement que pour les commandes "<-" et "->" de EDITER, et les positions qu'ils occupent sont vérifiées et replacées dans les intervalles valides lors du démarrage.

Lorsqu'un autre programme en assembleur est utilisé simultanément au Disk Manager, les adresses ci-dessus risquent d'être modifiées par ce programme. Elle doivent donc être restaurées à leur valeur propres avec, selon le cas, la commande AFFECTER, pour définir avec le premier buffer le drive et le slot, ou EDITER pour définir avec lui le second buffer et le numéro de buffer.

AFFECTER est la commande d'initialisation de la partie accès disque du Disk Manager : elle réserve un buffer, puis spécifie optionnellement le disque et le slot à connecter. L'initialisation peut aussi se faire par la commande LIRCARTE si elle spécifie un buffer; si c'est le cas, le slot 6 et le drive 1 seront adressés par défaut.

SYNTAXE DES COMMANDES

Chaque commande comporte une syntaxe obligatoire et des paramètres optionnels.

Les commandes utilisant des noms de programme permettent toutes les possibilités de spécification par des caractères génériques dits "wildcard characters". Ainsi :

```
programme
=rarme
prog=
p=og=me
=ro=mm=
=p=r=o=
=
```

pourront-ils tous sélectionner "programme".

Le point d'interrogation peut également servir de caractère générique.

Les paramètres sont placés entre parenthèses avec la syntaxe :

& COMMANDE (, ,)

Ces paramètres sont des paramètres d'entrée spécifiés par l'utilisateur, ou des paramètres de sortie résultant de l'exécution de la commande, ou - dans le cas des commandes de nom - des paramètres mixtes (d'entrée/sortie) : le nom (en entier ou avec wildcard) est

alors affecté à une variable de chaîne et on retrouve après l'exécution le nom exact du programme dans cette même variable.

Par convention, les paramètres d'entrée seront écrits en majuscules dans le texte, les paramètres de sortie en minuscules. Les paramètres sont séparés par une virgule; quand une commande comporte des paramètres numériques des deux types, les paramètres de sortie sont toujours placés en fin de commande et séparés des autres soit par une parenthèse droite, soit par une instruction Basic (LET, par exemple).

Les paramètres optionnels sont toujours placés en fin de commande, ou avant les paramètres de sortie si ce sont des paramètres d'entrée. Quand il y a plusieurs paramètres optionnels, c'est toujours le(s) dernier(s) de l'instruction qui est (sont) considéré(s) comme omis.

Exemple:

& CARTE (PST,SEC,BUF LET x)

où PST et SEC sont les paramètres d'entrée obligatoires. BUF est un paramètre d'entrée optionnel et x un paramètre de sortie. Ce dernier est également facultatif : il peut être omis, auquel cas le résultat est affiché à l'écran au lieu d'être affecté à x.

Le paramètre BUF, optionnel, peut être omis même si "LET x" est présent, car LET sert à distinguer x comme paramètre de sortie, et il ne peut être pris pour BUF par l'interpréteur.

Aucune commande ne peut être passée si un buffer n'a pas été défini une fois auparavant et n'est pas non plus spécifié dans la commande. Le message d'erreur "buffer non spécifié" stoppe l'exécution du programme.

Les buffers et HIMEM

L'emplacement de DOM et sa raison d'être privilégient l'emploi pour son propre usage des buffers réservés par le DOS. Deux de ces buffers, soit 1190 octets, offrent au Disk Manager l'emplacement nécessaire pour 4 buffers de 256 octets.

Un seul de ses buffers suffit au DOS pour le chargement du programme ou la gestion d'un fichier unique.

Peuvent également être utilisés :

- le buffer d'entrée à l'adresse 512
- le buffer de chargement du catalogue en 46011
- le buffer de chargement de la VTOC en 46267

Ces deux derniers buffers sont effacés par chaque opération normale d'accès au disque; c'est d'ailleurs la raison pour laquelle il n'ont pas été utilisés de façon interne par le Disk Manager. Le buffer d'entrée est pour sa part modifié par les instructions INPUT et à chaque entrée en mode direct. Aussi ces derniers buffers sont-ils d'un emploi assez peu recommandé.

L'adresse de départ d'un buffer, si l'on veut l'afficher avec DE (ce n'est pas le cas avec DOM seul) doit être un multiple de 256. On peut

créer dans les buffers du DOS quatre buffers dont les adresses sont les suivantes : 38400, 38656, 38912 et 39168.

Dans le reste de la mémoire, il est possible de réserver autant de buffers que nécessaire pour l'exécution d'une application; cette réservation est cependant purement dynamique, et leur emplacement peut être effacé de façon externe par le programme utilisant DOM.

En outre, le buffer utilisé par toute commande du Disk Manager est toujours le dernier buffer défini soit par une commande d'initialisation, soit par la dernière commande exécutée.

MAXFILES 1, suivi de CALL 36608 pour replacer HIMEM au-dessous du Disk Manager, permet au besoin de protéger ces buffers contre l'utilisation par le DOS. Ce dernier produira le message "BUFFER NON AVAILABLE" s'il ne trouve pas l'espace dont il a besoin.

MAXFILES 3 restitue la valeur par défaut. MAXFILES ne peut avoir une valeur supérieure à 3 sans effacer le Disk Manager.

Le sommet de la mémoire se situe donc en 36608 (#8F00) quand DOM est chargé seul. Il se situe en 34048 (#8500) pour protéger également DE.

L'utilisation de la mémoire pour charger une série de secteurs par programme nécessite une zone protégée où les variables de ce programme se localiseront. Deux procédés sont possibles :

- abaisser le sommet par HIMEM au début du programme, les buffers étant situés entre l'adresse de HIMEM et celle de DE (38048) ou de DOM (36608);
- utiliser les pointeurs de mémoire libre de l'Applesoft :
(PEEK(110)+1)*256 = adresse de début de mémoire libre
(PEEK(112)-1)*256 = sommet de la mémoire libre, utilisable pour des buffers

IV - Commandes de lecture/écriture

& LIRE (P,S)

Charge le secteur S de la piste P.

Option : & LIRE (P,S,BUF) permet d'affecter un buffer.

& ECRIRE (P,S)

Ecrit la page mémoire sur le secteur S de la piste P.

Option : & ECRIRE (P,S,BUF).

& LIRIND (x)

Affecte à x la valeur 1 si la dernière opération a engendré une erreur disque. Retourne 0 autrement.

& AFFECTER (BUF)

& AFFECTER (BUF,DR)

& AFFECTER (BUF,DRIVE,SLOT)

Spécifient l'adresse du buffer pour les opérations ultérieures éventuellement avec l'indication du drive et du slot. Par défaut, drive 1, slot 6.

LIRE

La commande de base LIRE charge tout secteur d'une disquette formatée en DOS 3.3 dans le dernier buffer affecté, ou le cas échéant dans le buffer désigné par la commande.

Si elle a été préalablement initialisée, la piste 36 peut être adressée. Le premier paramètre doit donc être compris entre 0 et 36 tandis que le second est compris entre 0 et 15.

L'adresse du buffer peut correspondre à toute adresse comprise entre 0 et 65280. Plus pratiquement, elle se tiendra dans la mémoire utilisateur, entre les adresses 2048 et 36608, ou dans les zones disponibles mentionnées dans le chapitre précédent.

En cas d'impossibilité de lecture disque, un indicateur visible par la commande LIRIND sera positionné, le programme Applesoft en cours ne sera pas interrompu et le message "ERREUR EN LECTURE" sera affiché.

ECRIRE

La commande ECRIRE copie le contenu du buffer sur un secteur désigné; ses caractéristiques sont identiques à celles de la fonction LIRE.

Le message "ERREUR EN ECRITURE" apparaîtra si la disquette n'est pas formatée en DOS 3.3 ou si le secteur est mauvais, le message "ECRITURE PROTEGEE" si la protection d'écriture est posée.

LIRIND

Toute erreur disque entraîne l'affichage d'un message, et le programme en cours n'étant pas interrompu, une instruction est nécessaire pour relever l'occurrence d'une erreur. La commande LIRIND doit donc suivre la commande dont on veut tester le résultat. Elle doit même la suivre immédiatement, sinon la commande suivante du Disk Manager remettra l'indicateur d'erreur à 0 si elle se déroule normalement.

L'utilitaire SCANNER inclus sur la disquette du Disk Manager permet de vérifier l'état d'une disquette et de marquer les mauvais secteurs éventuels. Les disquettes protégées ne peuvent évidemment pas être modifiées par le Disk Manager.

AFFECTER

Cette commande initialise le Disk Manager; elle est facultative mais il est préférable de l'utiliser en premier lieu. Dans le format intermédiaire, avec drive spécifié, avec slot spécifié ou non, elle remet le système à zéro : toute opération sur la carte d'occupation nécessitera que la carte soit rechargée à nouveau. Si une carte est en cours de mise à jour, elle doit donc être réécrite sur la disquette avant que la commande AFFECTER ne soit utilisée.

Aucune des commandes du Disk Manager ne peut être exécutée sans qu'un buffer n'ait été désigné. Cependant, AFFECTER est la seule pour laquelle le paramètre "buffer" est obligatoire.

La plupart des commandes permettent de spécifier le buffer en option, mais ce n'est pas le cas de LIRNOM, ECRNOM, aux paramètres très nombreux. LIRADR, ESPACE ne possèdent pas non plus cette option.

AFFECTER doit donc être employé chaque fois que l'on utilise ces commandes après emploi d'un autre buffer, notamment pour la mise à jour de la carte d'occupation.

V - Modification d'un secteur

```
& EDITER (BUFFER, NUMERO, BUFFER2)
& EDITER (BUFFER)
& EDITER
```

Edite le secteur chargé dans le buffer, ou simplement une page mémoire. Cette commande comporte elle-même de nombreuses fonctions. Taper "?" pour en obtenir la liste pendant le fonctionnement du programme.

On peut spécifier deux buffers et le numéro de celui qui sera affiché au départ, 1 ou 2.

Principes généraux

Bien que constituant seulement une partie du Disk Manager, cette commande correspond à un programme complet avec toutes les fonctions d'un utilitaire d'accès-disque. Elle se particularise par un emploi conçu le plus simple possible, et accessible au néophyte, tout en disposant de possibilités assez étendues pour satisfaire un utilisateur expérimenté. En la comparant aux utilitaires commerciaux spécialisés, on verra qu'elle ne possède pas les fonctions composées qui ont été intégrées à ceux-ci, telles que celles qui manipulent la VTOC, mais ces fonctions se réalisent aisément avec les autres commandes du Disk Manager.

L'accès aux paramètres principaux, situés dans les adresses 24 à 31, permet de programmer cette commande et de réaliser des séquences d'opérations interactives.

Usage interactif

Toutes les valeurs données à l'éditeur peuvent être entrées soit en décimal, soit en hexadécimal si elles sont alors précédées de "\$". L'affichage des numéros de piste et secteur peut également se faire en décimal ou en hexadécimal, la touche H permettant de basculer de l'un à l'autre.

On ne doit pas appuyer sur RETURN pour annuler une commande mais frapper un point, puis faire RETURN. Si l'on entre RETURN en réponse à une question, cela valide la valeur par défaut; pour les numéros de piste et secteur, les valeurs par défaut sont celles qui apparaissent en en-tête. Pour le désassemblage, la position du curseur sur la page affichée est la position de début par défaut.

La commande démarre par "& EDITER" dans l'environnement complet ou par "CALL 10" si DE est seul chargé.

1. en-tête

Dès le départ, on voit apparaître deux lignes de référence et une page de mémoire dans un cadre en inverse qui donne les coordonnées de chaque octet de cette page.

Celle-ci est présentée en codes ASCII, mais on peut basculer avec la représentation en hexadécimal en appuyant sur la barre d'espacement. Quand on a placé le curseur sur un caractère, il suffit d'appuyer sur la barre pour obtenir instantanément son équivalent hexadécimal, et inversement pour voir la conversion en ASCII d'un code hexadécimal.

L'en-tête donne les indications suivantes, dont la signification sera détaillée plus loin.

PST : numéro de piste pointée
SEC : numéro de secteur pointé
DRV : 1 ou 2, numéro du drive branché
BUF : 1 ou 2, numéro du buffer affiché
XV : OUI/non. Mode disque ou mémoire
PAGE : Adresse de départ de la page mémoire affichée,
contenant un secteur disque ou non
F : oui/non. Filtre actif ou non
I : oui/non. Ecriture en inverse ou non
C : oui/non. Ecriture en contrôle ou non
LOCK/UNLOCK: buffer verrouillé ou déverrouillé

Attention !

L'utilisateur ne possédant pas une sérieuse connaissance de l'Apple II doit vérifier que le verrou, spécifié par "LOCK" est toujours posé, sinon un secteur peut être chargé à tout endroit de la mémoire centrale, détruire la page zéro ou le DOS, entraînant des conséquences irréversibles. En cas de doute sur les manipulations que l'on a faites, il est prudent de re-booter le DOS avant tout usage du drive. Si aucun programme n'est présent en mémoire, les pages mémoire de 2048 (\$800) à (\$8400) peuvent être utilisées sans inconvénient.

Les buffers

Le buffer utilisé par défaut occupe les positions \$9600 à \$96FF (38400-38655). La commande EDITER permet d'adresser deux buffers de base; on bascule de l'un à l'autre avec la touche B. Le contenu du buffer sélectionné est affiché sur l'écran; comme on le verra plus loin, il est cependant possible de visualiser toute page de mémoire et de s'en servir comme buffer pour la lecture et l'écriture de secteurs.

Un même buffer peut être utilisé par DOM et l'éditeur, à condition que son adresse soit un multiple de 256, l'éditeur opérant sur des pages de mémoire.

La sélection de page ou secteur

L'écran place dans un cadre référence les 256 octets d'un secteur chargé dans une page mémoire. Il présente les secteurs de la disquette, soit par un déplacement séquentiel avec les touches -> et <-, soit par positionnement direct avec la touche L suivie des numéros de piste et secteur. On peut également balayer la mémoire centrale en débranchant l'accès disque (touche D).

Les commandes "<-" et "->" ne fonctionnent qu'en mode lecture, le fait que le drive soit branché ou non étant indiqué en haut à droite de l'écran par l'indication "drive:OUI" ou "drive:NON".

Toute page mémoire visualisée peut être modifiée et enregistrée sur un secteur donné en déverrouillant les buffers avec la touche "=". Une page mémoire se sélectionne par la commande P, mais elle arrondit si nécessaire au multiple de 256 juste inférieur à la valeur donnée. Cette valeur, comme les valeurs entrées pour chaque fonction, peut être décimale ou hexadécimale.

La sélection d'un secteur à lire est commandée par la touche L; l'écran affiche alors :

"LIRE PISTE --- SECTEUR --"

Le programme attend que l'on indique un numéro de piste (en décimal ou hexadécimal) et un numéro de secteur. On peut appuyer sur RETURN si la piste est la même que celle qui est affichée en référence, de même pour le numéro de secteur.

L'écriture du secteur s'opère selon le même principe : on appuie sur E, et on voit s'afficher :

"ECRIRE PISTE --- SECTEUR --"

L'accès disque, en lecture comme en écriture, se fait toujours par l'intermédiaire d'un des deux buffers réservés, celui qui est sélectionné et indiqué en ligne 1.

Afin de donner de plus larges possibilités aux utilisateurs expérimentés, le déverrouillage de cette protection est accessible par la touche "=", et il permet de charger le secteur dans la page mémoire affichée quelle qu'elle soit, et c'est également la page affichée qui est copiée sur le disque.

Lorsqu'aucun programme ne cohabite avec la commande EDITER, la mémoire est libre entre les adresses 2048 (\$800) et 33792 (\$8400) incluses. Si un programme est présent en mémoire, une zone protégée par HIMEM permet d'utiliser la mémoire sans interférer avec les variables du programme.

La modification d'un secteur

Un exemple présenté plus loin démontre la facilité d'édition d'un secteur. Les touches I,J,K,M permettent de placer le curseur sur la position voulue. On revient au début du buffer par A, et on saute à la fin par Z.

La touche RETURN permet de se placer en mode écriture, et de sortir de ce mode. Tout caractère au clavier peut être placé sur l'écran ASCII en donnant les deux digits de son code hexadécimal sur écran hexadécimal.

Les codes ASCII entrés correspondent aux caractères affichés en mode normal. Deux autres options existent :

- les caractères de contrôles : ils sont entrés comme les codes normaux après branchement de l'option C:oui.
- les caractères en inverse : ils sont entrés après branchement de l'option I:oui.

La touche C et la touche V, respectivement, basculent ces deux options.

La réalisation d'un en-tête de disquette comme celui du Disk Manager permet d'illustrer ces notions... Prenons une disquette vierge et inscrivons-y sept entrées sur le catalogue. On peut utiliser les commandes suivantes : POKE 40993,173 : FOR I = 1 TO 7 : PRINT CHR\$(4) "OPEN N" I : NEXT : PRINT CHR\$(4) "CLOSE"

Ensuite, chargeons et affichons sur l'écran, à l'aide de la commande LIRE, le premier secteur du directory, piste 17, secteur 15.

Après le positionnement du curseur sur le premier caractère du premier nom, en position 14, tapons C puis RETURN pour l'entrée de codes de contrôle, et la lettre "H" sur les 7 premières positions du nom. C'est le backspace qui ramènera le nom à gauche de l'écran lors de l'affichage.

Ensuite RETURN, puis C pour revenir au mode normal et V avec RETURN pour écrire en inverse les lettres suivantes sur les 23 positions restantes, en complétant par des espaces.

Ceci est répété sur les 7 noms; on tape alors E pour écrire, puis deux fois RETURN : le secteur est maintenant réécrit.

"/", pour obtenir le CATALOG, permet de se rendre compte de suite du résultat.

Le filtre

En mode filtre le programme n'affiche que des codes en caractères normaux. Les codes de contrôle sont remplacés par des tirets.

Le désassemblage

On peut désassembler un programme directement sur la disquette avec la commande "S". Le désassemblage comporte lui-même plusieurs commandes. Il démarre en affichant:

"DEPART:"

pour obtenir l'adresse de départ que l'on peut fournir en décimal ou en hexadécimal. Si l'on appuie sur RETURN, une adresse par défaut est prise, qui correspond à la position pointée par le curseur dans le buffer. Vingt lignes sont alors désassemblées et l'on a le choix entre plusieurs options :

- la barre d'espace pour désassembler ligne par ligne
- RETURN pour désassembler les 20 lignes suivantes
- D pour donner une nouvelle adresse de départ
- ESCAPE pour retourner au cadre d'édition

Cette commande permet de retrouver des programmes quand le directory est accidentellement détruit, en recherchant les TSL de programmes sur la disquette, en identifiant le programme correspondant par désassemblage des premières lignes, puis en reconstituant un secteur de directory.

En effet, les TSL se reconnaissent à leur format, décrit plus haut dans la section sur l'analyse d'une disquette. Le premier secteur d'un programme est spécifié en positions 12 (numéro de piste) et 13 (numéro de secteur).

Autres commandes

La liste des commandes s'obtient en appuyant sur "?". Cette liste donne le mode d'emploi du clavier, mais est très succincte, le présent manuel donnant une description détaillée de chaque commande.

Le drive par défaut est le drive 1, sauf si l'on a branché le second avec l'instruction AFFECTE ou par un POKE en 26. Le drive 1 ou 2 peut être branché directement en appuyant sur les touches "1" ou "2".

La récapitulation qui suit donne la liste de toutes les commandes.

RECAPITULATION DES COMMANDES D'EDITION

- A : place le curseur en début de page
- B : changement de buffer
- C : bascule pour entrée de caractères de contrôle
- D : bascule entre modes disque et mémoire pour les commandes <- et ->
- E : écriture du buffer sur le secteur pointé ou sur un secteur spécifié
- F : bascule qui pose ou enlève le filtre
- H : bascule décimal-hexadécimal pour les numéros de piste et secteur

I,J,K,M : déplacement du curseur (plus A et Z)
 L : lecture d'un secteur
 : bascule le cadre entre les modes normal et inverse
 P : sélection d'une page de mémoire, à visualiser ou comme buffer
 V : bascule pour entrée de caractères en inverse
 Z : place le curseur en fin de page
 = : bascule du verrou qui oblige, quand il est posé, le chargement
 d'un secteur dans le buffer désigné, 1 ou 2
 <- et -> : En mode mémoire : affiche la page précédente ou la
 suivante,
 respectivement. En mode disque : lit le secteur précédent ou le
 suivant, sur la même page
 RETURN : bascule début/fin du mode entrée de caractères
 ESC : sort de la commande EDITER, retour au Basic ou au programme
 appelant
 ? : liste des commandes
 1/2 : spécification du drive 1 ou 2
 / : catalogue du drive spécifié
 * : désassemblage du buffer ou d'une zone donnée
 D : nouveau départ
 Barre : une nouvelle ligne
 RETURN : 20 nouvelles lignes
 ESC : retour au cadre
 : lorsque l'on a commandé une fonction qui requiert une adresse,
 taper "." puis RETURN pour annuler la commande. RETURN seul fait
 prendre une adresse par défaut.

VI - Commandes de carte et de nom

& LIRCARTE (BUF) ou & LIRCARTE

Chargement de la carte d'occupation de la disquette dans le buffer.

& ECRCARTE (BUF) ou & ECRCARTE

Ecriture de la carte contenue dans le buffer.

& CARTE (P,S LET x)

Attribue à x la valeur 1 si le secteur S de la piste P est indiqué comme occupé, la valeur 0 sinon.

La commande LIRCARTE est exécutée automatiquement si la carte n'a pas été chargée depuis la dernière commande AFFECTER.

Option écran : & CARTE (P,S) affiche la valeur 1 ou 0 au lieu de l'affecter à une variable.

Option buffer : &CARTE (P,S,BUF LET x) ou & CARTE(P,S,BUF).

& INSCARTE (P,S)

Marque sur la carte d'occupation le secteur S de la piste P comme occupé.

Option buffer existante. Peut forcer LIRCARTE.

& SUPCARTE (P,S)

Marque sur la carte d'occupation le secteur S de la piste P comme libre.

Option buffer existante. Peut forcer LIRCARTE.

& LIRNOM (A\$,p,s,t,l)

Cherche sur le catalogue le nom de programme A\$, passe le numéro de piste de sa TSL dans la variable p, le numéro de secteur dans s, le type de programme ou fichier dans t et la longueur dans l.

Option numéro : & LIRNOM (N,A\$,p,s,t,l). La recherche s'effectue alors à partir du numéro de programme N.

Option wildcard : le caractère "=" remplace le début, la fin, une partie ou plusieurs du nom du programme, et le nom réel sera chargé dans A\$.

Option CHRGNOM : le nom du programme chargé n'est pas converti en format Applesoft et il a toujours une longueur de 30 caractères complétée à droite par des espaces.

& ECRNOM (A\$,P,S,T,L)

Ajoute ou remplace le nom A\$ sur le catalogue; le numéro de piste de la TSL est P, le numéro de secteur S, le type T et la longueur L.

Option SAUVNOM : écrit le nom tel que contenu dans la variable AS.
Options numéro et wildcard comme pour & LIRNOM.

& ESPACE(x)

Affecte à x l'espace libre sur la disquette à partir de la carte d'occupation. Même remarque que pour CARTE.

Option écran : & ESPACE affiche l'espace libre au lieu de l'affecter à x.

& LIRADR (AS,dep,lng)

Cherche sur la catalogue le programme binaire AS, affecte à dep l'adresse de départ en mémoire, à lng sa longueur.

Option écran : & LIRADR (AS) affiche l'adresse de départ et la longueur.

Options numéro et wildcard comme pour & LIRNOM.

VI-1 Allocation et libération de secteurs

A - LIRCARTE

La carte d'occupation de la disquette est chargée dans le buffer désigné ou dans le dernier buffer utilisé. Dans l'analyse de la disquette, au chapitre II, nous indiquons la composition exacte de cette carte, dont on peut lire le contenu dans le buffer par la commande PEEK ou le modifier par POKE.

Cette commande est facultative et permet surtout dans certains cas de spécifier un buffer pour la carte, car son chargement est de toute façon réalisé à la première occurrence d'une des instructions de gestion de la carte.

Comme AFFECTER, elle initialise le système et, si la commande AFFECTER n'a pas été exécutée, met en ligne par défaut le drive 1 et le slot 6.

B - ECRCARTE

Recopie la carte d'occupation sur la disquette. Le buffer recopié est soit le buffer spécifié dans la commande, soit le dernier buffer utilisé. Il est nécessaire en général de spécifier le buffer dans toutes les commandes de carte, car ces commandes s'utilisent corrélativement à d'autres commandes d'accès disque.

ECRCARTE n'oblige pas les commandes de gestion de la carte à la recharger de nouveau.

C - CARTE, INSCARTE, SUPCARTE

CARTE teste si le secteur spécifié est libre, renvoie 0 si c'est le cas, 1 autrement.

INSCARTE marque ce secteur comme non utilisable.

SUPCARTE le marque libre.

Ces trois commandes permettent de spécifier le buffer où LIRCARTE a chargé la carte, après une autre commande. Elles la chargeront automatiquement en mémoire, après chargement de DOM, ou changement de drive par AFFECTER.

D - ESPACE

Comme CARTE, il s'agit d'une commande dont le résultat peut être affiché directement quand on ne spécifie pas de variable de sortie. En affichage écran, la syntaxe est :

& ESPACE ou :

& ESPACE (x) pour affecter à x l'espace libre.

Il n'a pas de retour à la ligne comme cela se passe avec la commande PRINT quand elle n'est pas suivie de ";" ou ",,".

ESPACE force l'exécution de LIRCARTE si la carte d'occupation n'a pas été chargée préalablement mais, si aucun buffer n'a été spécifié, cela provoque le message "BUFFER NON SPECIFIE".

VI-2 Commandes de nom

Une syntaxe et des options communes définissent ces commandes :

- le nom du programme se spécifie dans une variable, appelée ici AS, ou entre guillemets, pour notamment l'accès en mode direct
- la variable de numéro N permet de pointer sur une position du catalogue donnée
- quatre paramètres sont nécessaires pour chaque commande d'écriture; deux seulement sont obligatoires pour chaque commande de lecture, deux autres enfin sont optionnels;
- le caractère générique est valide pour toute instruction, que le nom soit affecté à une variable ou placé entre guillemets.

A - LIRNOM, ECRNOM, CHRGNOM, SAUVNOM

Le format complet de la commande LIRNOM est le suivant :

& LIRNOM (N,AS,P,S,T,L)

où : N signifie "à partir du programme numéro N sur la disquette"

AS est le nom du programme (exact ou avec wildcards)

P sera le numéro de piste de la TSL, si le nom est trouvé

S sera le numéro de secteur de la TSL
T contiendra après la recherche le code de type du programme
L contiendra la longueur affichée sur le catalogue

Ces deux derniers paramètres, ainsi que l'option N, peuvent être omis.

C'est la variable P qui permettra de savoir si le programme a été trouvé ou non. Dans le second cas, elle aura la valeur 0, les autres variables ayant alors des valeurs aléatoires.

S'il a été trouvé sur la disquette, le nom exact du programme sera retourné dans la variable AS, qui peut contenir au départ un schéma avec wildcards ou le code "=" seul.

Ceci permet également de vérifier si le programme a été trouvé ou non et d'afficher le nom; s'il n'a pas été trouvé, le schéma original demeurera dans la variable AS. C'est en fait le seul moyen de le savoir pour la commande LIRADR, quand elle est employée dans un programme. Il est plus facile avec LIRNOM de tester si la variable P contient zéro.

Comme la variable de nom, la variable N de numéro est une variable mixte qui fait démarrer la recherche dans le catalogue à partir de la position donnée, et renvoie après exécution le numéro de la position où le programme a été trouvé.

Cette recherche compare le nom donné et le nom trouvé sur les 29 premières positions.

CHARGNOM est identique à LIRNOM quand aux paramètres, options et fonctionnement. La première différence tient au format du nom lu. Après LIRNOM, on aura dans AS un nom de longueur variable, avec le bit 7 à 0 comme c'est le cas pour toutes les variables de chaîne Applesoft. CHARGNOM conservera pour sa part le format du nom sur le catalogue, en 30 caractères, complétés si nécessaire par des espaces, et avec le bit 7 à 1. La seconde différence est que CHARGNOM prend en compte tous les programmes présents sur le DIRECTORY, y compris ceux qui ont fait l'objet d'un DELETE.

Les tableaux numériques ne peuvent être employés comme paramètres, pour les variables P, S, T et L, dans les commandes de lecture de noms, alors que cela est autorisé pour les commandes d'écriture. La raison en est que la procédure utilisée pour affecter une valeur aux variables de sortie numérique utilise une routine de l'Applesoft qui crée la variable dans la table des variables, si elle ne la trouve pas. Or cette procédure fonctionne différemment selon le type de variable utilisé. Les variables entières ne sont pas valides non plus.

ECRNOM place sur la disquette un nom de programme en format disquette, même si le nom est en format Applesoft dans AS. L'option N fonctionne comme avec les commandes de lecture, et fait que l'on ne prend en compte le catalogue qu'à partir du programme numéro N.

SAUVNOM écrit le nom tel qu'il est donné, quel que soit son format

dans AS (dans la limite de 30 caractères). S'il est en format Applesoft, on le verra apparaître en clignotant sur le catalogue. SAUVNOM prend également en compte les programmes DELETés, et cela est vrai aussi pour la variable de numéro qui prendra pour valeur leur nombre ajouté à celui des autres programmes. Elle permet de replacer un programme à sa place dans le catalogue, après effacement, tandis qu'il est affiché en fin de catalogue, quand son nom n'y figure pas déjà.

Les paramètres P, S, T et L sont tous nécessaires pour les commandes d'écriture. L'autre différence est que, si les wildcards peuvent s'employer, le nom éventuellement trouvé n'est pas affecté à AS; enfin, s'il n'est pas trouvé, le schéma avec wildcard sera pris comme nom et ajouté en fin de catalogue.

B - LIRADR

Cette commande a comme ESPACE une syntaxe différente selon l'emploi en mode direct ou indirect.

& LIRADR(AS) ou & LIRADR ("NOM")
suffisent pour lire les adresses d'un programme binaire en mode direct. Elles sont affichées en hexadécimal et décimal.

En mode indirect,

& LIRADR(AS,d,l)
affecte l'adresse de départ à d, la longueur à l.

Les options de LIRNOM restent valides ici.

VII - Premières applications

VII-1 Applications simples

LE CATALOGUE DANS UN TABLEAU

Avec l'option "numéro", il est facile de charger le catalogue dans un tableau de chaînes de caractères, ou plus exactement dans des tableaux, si l'on souhaite aussi charger le type et la longueur des programmes.

```
10 FOR N= 1 TO 105
20 A$= "="
30 & LIRNOM(N,A$(N),P,S,T,L)
40 IF NOT P THEN 100
50 Tx(N)=T
60 Lx(N)=L
70 Px(N)=P
80 Sx(N)=S
90 NEXT N
100 REM suite du traitement
```

La variable A\$ contient le wildcard "=" ou un nom comprenant des wildcard, avant que la commande LIRNOM ne soit passée; elle contiendra le nom exact, après exécution. N permet d'effectuer la recherche à partir du numéro du dernier nom lu. CHARGNOM permettrait de charger aussi les noms des programmes DELETés.

REECRITURE DU CATALOGUE

On peut vouloir récrire le catalogue, par exemple pour le purger des programmes DELETés. Le catalogue sera alors chargé par la procédure précédente, puis les 15 secteurs réservés seront effacés comme dans la routine ci-dessous :

```
10 ZERO = 0
```

```
Localisation du catalogue :
20 & LIRCARTE(BUF)
```

```
Adresse du secteur suivant :
30 P = PEEK(BUF+1)
40 S = PEEK(BUF+2)
50 IF S=0 THEN 120
60 & LIRE(P,S,BUF)
```

```
Effacement des noms :
70 FOR I = 11 TO 255
80 POKE I,ZERO
90 NEXT
100 & ECRIRE(P,S,BUF)
110 GOTO 30
```

120 REM suite du programme

Le numéro de piste et de secteur du début du catalogue est lu dans la carte d'occupation. Le numéro de secteur suivant est lu en position 2 de chaque secteur, le code "0" signalant que l'on a chargé le dernier. On efface les zones réservées aux noms et descriptions de programmes de la position 11 à la position 255. Réécrire le catalogue s'effectue par une procédure similaire à celle de la lecture.

```
10 FOR N=1 TO 105
20 IF P*(N)=0 THEN 50
30 & ECRNOM (A$(N),P*(N),S*(N), T*(N),L*(N))
40 NEXT N
50 REM suite du traitement
```

LOCK/UNLOCK EN SERIE

Les opérations en série s'effectuent avec les procédures de lecture et écriture précédentes. La formule:

TYPE= TYPE-128*(TYPE>127)+128*X

fait qu'un programme sera déverrouillé ou verrouillé quel que soit son état antérieur, si l'on attribue à X la valeur 0 pour le déverrouillage et 1 pour le verrouillage.

DOS ABSENT

Lorsque l'on a récupéré, sur une disquette destinée au seul stockage, la place occupée par le DOS, le boot de cette disquette entraîne un retour au moniteur que l'on peut remplacer plus agréablement par l'affichage d'un message.

Un petit programme en assembleur affichera ce message :

```
ORG $801
LDY $2B ;adresse du slot
LDA $C088,Y ;stoppe le moteur
JSR $FC58 ;home
LDA $0 ;provoque un reboot quand
STA $3F3 ;on appuie sur Reset
LDY $0 ;position 0 sur texte
boucle LDA texte,y
BEQ boucle ;test fin de message
JSR $FDED ;affiche un code
INY ;position suivante
BNE boucle
texte ASC "DOS ABSENT - BOOTER UNE AUTRE DISQUETTE"
```

Ce programme (dénommé "MESSAGE" sur la disquette du Disk Manager) étant chargé (par BLOAD), on charge ensuite DE par BRUN s'il n'est pas déjà en mémoire; sinon, on le lance par CALL 10, ou & EDITER.

Mettre en mode UNLOCK avec "=", taper P pour sélectionner la page puis 2048 (ou \$800) pour le numéro de page.

La recopie du secteur s'obtient en insérant la disquette à mettre à jour, et en tapant E pour écrire avec, comme paramètres, piste 0 et secteur 0.

Mais, auparavant, le texte qui accompagne le programme assembleur et qui se détache clairement en mode filtre (taper "F"), peut être modifié ou étendu; il peut même occuper tout le reste du buffer. Le code RETURN permet de placer un retour à la ligne et peut être entré en mode hexadécimal : taper 8D ou, en mode contrôle, taper "C" <RETURN>, puis "M" <RETURN> et enfin "C" pour revenir au mode normal.

L'utilisateur peut placer un message de son choix comme "DOS absent, booter une autre disquette" ou un message plus amusant ... "Zone de stockage!" ou "Aie ! Pas de DOS sur cette disquette !".

CLASSEMENT DU CATALOGUE

Bien que cela semble a priori une opération dangereuse, dont on pourrait préférer s'abstenir pour ne pas risquer de perdre une disquette, le classement alphabétique d'un catalogue peut se faire en toute sécurité et s'opère simplement en chargeant le catalogue dans des tableaux et en classant ces tableaux.

Le tri doit évidemment échanger les paramètres piste, secteur, type et longueur chaque fois qu'il échange deux noms de programmes ! Ce tri pourrait aussi en fait s'opérer sur la base du type ou la longueur, si l'on préfère. Il aurait alors pour résultat de classer par longueur croissante ou décroissante, à partir des codes correspondant aux types, identifiés par le DOS comme suit :

0 = fichier
1 = Integer
2 = Applesoft
4 = binaire

En outre, les programmes verrouillés, pour lesquels ces valeurs sont augmentées de 128, se trouveraient automatiquement placés à la fin.

Pour éviter cet inconvénient, on renumérottera les types en fonction de l'ordre voulu, mais en utilisant une numérotation qui permette toutefois de retrouver les codes originaux après le tri et avant la réécriture du catalogue !

Exemple - Pour classer dans l'ordre suivant : Applesoft, binaire, Integer, fichier :

0 deviendra 3, 128 sera 131
1 deviendra 2, 129 sera 130
2 deviendra 0, 130 sera 128
4 deviendra 1, 132 sera 129.

La comparaison dans le classement ne se fera pas sur la valeur brute mais sur la valeur réduite à celle du programme non verrouillé par la formule :

$TYPE-128*(TYPE>127)$

Le catalogue peut aussi être réorganisé manuellement, ce qui se réduit à une procédure de gestion de tableau dont un algorithme est proposé dans l'éditeur de catalogues.

LA CARTE D'OCCUPATION SUR L'ECRAN

L'instruction CARTE affectant à x l'indice d'occupation d'un secteur, on charge la carte d'une disquette, et on teste l'état de chaque secteur :

```
10 FOR I= 0 TO 34
20 FOR J= 0 TO 15
30 & CARTE (I,J LET x)
40 IF X=1 THEN PRINT"*";
50 IF X=0 THEN PRINT " ";
60 NEXT J
70 PRINT
80 NEXT I
```

VII-2 Applications plus élaborées

FIX, CORRECTION DE LA LONGUEUR D'UN PROGRAMME

Les éléments nécessaires pour faire cette vérification sont :

- la/les TSL du programme
- la description du programme dans le catalogue : nom, coordonnées de TSL, et longueur

La commande LIRNOM nous donnera les coordonnées de la TSL et LIRE la chargera dans le même buffer.

Des positions 12 à 254, un compteur est incrémenté tant que la valeur lue est différente de zéro, ces positions étant lues avec un pas de deux, une paire d'octets représentant les coordonnées d'un unique secteur.

La position 1 est testée pour l'existence d'une autre TSL. Si elle est différente de zéro, on revient en début de boucle et on donne à la commande LIRE les valeurs trouvées en position 1 et 2.

Quand la position 1 correspond à zéro, on peut mettre à jour le DIRECTORY avec la commande ECRNOM, qui reprend les paramètres lus dans la commande LIRNOM utilisée au départ, sauf le paramètre longueur, qui reçoit la valeur du compteur.

Trois commandes du Disk Manager suffisent donc pour cette application !

La procédure de lecture de TSL correspond au programme général ci-dessous. La sortie de boucle se fait dès qu'un numéro de piste est nul pour les programmes; elle s'effectue en fin de TSL pour les fichiers, puisque les fichiers à accès direct n'ont pas de TSL linéaire.

Lecture du nom :

```
10 NUM=0 : BUF=38400
20 & LIRNOM(AS,p,s,t,l)
30 t = t-128*(t>127) : REM 0 si fichier
40 B=p : C=s
```

```

Boucle de lecture TSL multiples :
50 & LIRE(B,C,BUF)
   FOR I=BUF+12 TO BUF+254 STEP 2
70 P2 = PEEK(I)
80 IF NOT P2 AND NOT t THEN 130
90 IF NOT P2 THEN NEXT : GOTO 130
100 S2= PEEK(I+1)
110 REM traitement ici
    NUM=NUM+1 pour compter secteurs
120 NEXT I

```

```

Secteur suivant éventuellement :
130 B= PEEK(BUF+1)
140 IF B THEN C=PEEK(BUF+2) : GOTO 50
150 REM suite du traitement

```

On peut placer le traitement voulu aux lignes 110 et suivantes; dans notre exemple, un compteur décompte le nombre de secteurs occupés; dans la version intégrée au programme U-D, la carte d'occupation est en outre mise à jour, pour corriger les défauts éventuels d'imputation.

LE SCANNER

L'opération qui consiste à vérifier le bon état d'une disquette et protéger contre l'écriture les secteurs détériorés a fait l'objet de plusieurs programmes commerciaux. Elle nous est facile à réaliser avec le Disk Manager.

Elle fait intervenir la TSL de tous les programmes de la disquette et la carte d'occupation.

La commande LIRE est insérée dans une double boucle, de 0 à 34 pour les pistes et de 0 à 15 pour les secteurs; elle est associée à la commande LIRIND qui indique au programme si le secteur a pu être lu, ou si une erreur disque est survenue. Chaque secteur peut être lu et réécrit pour une double vérification.

Un mauvais secteur sera déclaré occupé sur la carte (commande INSCARTE) lorsque LIRIND répondra par la valeur 1, qui signale une lecture ou écriture impossible. Mais le traitement ne s'en tient pas là. Ce secteur détérioré peut être utilisé en ce moment pour la mémorisation d'un programme dont la lecture serait de ce fait impossible.

Alors, comme on l'a fait précédemment pour la procédure FIX, on lira les TSL de chaque programme pour éventuellement détecter l'affectation de ce secteur.

La question de sa destruction sera alors posée par le SCANNER et, en cas de réponse affirmative, la commande "DELETE" effacera ce programme.

Certains programmes commerciaux créent en outre un programme fictif baptisé "mauvais secteur" qui en reprend les coordonnées et permet de le signaler, ce que l'on peut ajouter avec ECRNOM, les numéros de

piète et secteurs de TSL étant les coordonnées du secteur détérioré.

ECRNOM, LIRNOM puis SAUVNOM permettraient de créer un nom en inverse.

47 SECTEURS EN PLUS

Les précédentes applications laissent deviner comment on utilisera la carte d'occupation pour libérer les secteurs occupés par le DOS, ceux des pistes 1 et 2, ou pour les déclarer occupés, sur cette carte. La piste 0 ne peut être récupérée sans modification du DOS lui-même, sauf pour y placer le catalogue, en mettant à jour les positions 1 et 2 de la carte d'occupation.

Des programmes ingénieux ont été publiés, qui récupèrent la place occupée par le DOS sur une disquette, mais ils laissent toutefois la responsabilité à l'utilisateur de décider si le DOS peut ou ne peut pas être supprimé. S'il a été supprimé antérieurement, on libère alors sur la carte d'occupation des pistes qui peuvent avoir déjà été réutilisées par des programmes, dont le destin sera ainsi scellé !

Nous savons, avec la procédure FIX, vérifier l'utilisation d'un secteur par un programme, et nous pouvons nous assurer qu'aucune TSL ne fait référence aux pistes 1 ou 2 avant de les libérer sur la carte, ou avant de restaurer ou recopier le DOS !

Sachant que le directory comporte 7 programmes par secteur et qu'il se trouve sur la piste 17, des secteurs 15 à 1 en descendant, on peut récupérer les secteurs 14, 13 ... à 1 de cette piste, si la disquette est destinée à ne stocker que quelques grands programmes.

Toutefois, les programmes n'occupent pas toujours des positions voisines sur le directory, des programmes DELETés pouvant s'intercaler entre deux noms affichés. Ayant vu plus haut comment procéder pour épurer le directory, nous pouvons éviter tout accident !

UNDELETE, RESTAURATION DE PROGRAMME

CHRGNOM permet de lire les paramètres d'un programme dont on donne le nom. ECRNOM permet de replacer un nom en normal, à la même place sur le catalogue. Le paramètre piste prendra la valeur ASCII du dernier caractère du nom pour remplacer le code 255 propre aux programmes DELETés. La procédure de lecture de TSL intervient encore ici. Elle est employée deux fois :

- avec CARTE pour vérifier qu'aucun secteur de la TSL n'est réutilisé
- avec INSCARTE ensuite pour mettre à jour la carte à partir de la TSL

RECONSTRUCTION DE LA CARTE D'OCCUPATION

La reconstruction de la carte d'occupation d'une disquette nécessite deux buffers réservés :

- au chargements successifs de toutes les TSL
- à l'emplacement de la carte que l'on va construire avant de la copier

Ce second buffer peut être entièrement mis à zéro avant toute ération, mais il faut mettre à 255 entre les positions 56 et 193 pour représenter les secteurs vides avant la mise à jour.

LIRNOM donnera les coordonnées des TSL de chaque programme et celles-ci seront chargées l'une après l'autre en enchainant comme précédemment les TSL multiples pour un même programme.

Les positions 12-13 à 254-255 seront lues deux par deux, la première valeur étant affectée à la variable piste, la seconde à la variable secteur, de la commande INSCARTE(PISTE,SECTEUR).

Une variable de test sera mise à 0 avant la boucle et mise à 1 dès qu'un numéro de piste sera inférieur à 3, pour la raison suivante : les pistes 1 et 2 qu'occupe le DOS devraient également être inscrites, mais rien ne prouve que le DOS n'est pas effacé sur la carte originale ! Ce test détecte la récupération des pistes affectées au DOS par un programme.

La carte contient aussi d'autres indications qu'il faut placer directement par POKE. Les adresses des POKES ci-dessous sont des adresses relatives, qu'il faut ajouter à celle du buffer utilisé. Les valeurs proposées ici sont des valeurs standard.

POKE 1,17 : numéro de piste du premier secteur du catalogue
POKE 2,15 : numéro de secteur du premier secteur du catalogue
POKE 3,3 : numéro de DOS
POKE 6,254 : volume
POKE 39,122 : nombre maximum de secteurs adressés par une TSL
POKE 48,18 : dernière piste où un secteur a été alloué
POKE 49,255 : sens de l'allocation : +1 ou -1 (donc 255)
POKE 52,35 : nombre de pistes par disquette (35 ou 36)
POKE 53,16 : nombre de secteurs par piste
POKE 54,0 : nombre d'octets par secteur, partie basse
POKE 55,1 : nombre d'octets par secteur, partie haute

On peut éventuellement définir POKE 192,X : allocation des secteurs 8 à 15 d'une 36ème piste et POKE 193,X : allocation des secteurs 0 à 7 de cette piste. Se reporter au chapitre relatif à la carte d'occupation dans l'analyse d'une disquette pour voir la façon dont opère l'allocation des secteurs.

Le numéro de la dernière piste allouée n'a pas de valeur objective. Il vaut mieux occuper les pistes de numéros les plus haut d'abord, l'idéal étant de placer les programmes les plus utilisés le plus près du directory.

On utilisera ici la piste 18 avec un sens d'allocation de -1 placé en position 49.

On peut à ce stade copier la VTOC mise à neuf par & ECRCARTE(BUF), où buf est l'adresse de départ du buffer utilisé.

COPIE PISTE A PISTE

Cette opération requiert une gestion de l'espace mémoire présent; les pointeurs 110 et 112 donnent le début et la fin de la mémoire

libre, en multiples de 256.

Toutes les variables doivent cependant avoir au moins une occurrence avant lecture de ces pointeurs, dont la valeur cessera d'être significative dès qu'une nouvelle variable modifie l'allocation mémoire. Aussi les initialise-t-on au début, même à 0.

```
10 P = -1 : S = 15 : FIN=34
20 PAGE = 256 : BUF = 0
30 LIRE = 1
40 IF LIRE THEN PRINT "SOURCE"
50 IF ECRIRE THEN PRINT "OBJET"
60 WAIT -16384,128 : POKE-16368,0
70 IF LIRE THEN P2=P : S2=S
80 PLAFOND = PEEK(112) * PAGE
90 BUF = PEEK(110) * PAGE
100 S = S2
110 P = P2
120 S = S + 1
130 IF S > 15 THEN S = 0 : P = P + 1
140 IF (P > FIN) AND LIRE THEN BUF = BUF - PAGE
150 IF (P > FIN) AND ECRIRE THEN END
160 BUF = BUF + PAGE
170 IF BUF = PLAFOND THEN LIRE = 1 - LIRE
    : ECRIRE = 1 - ECRIRE : GOTO 40
180 IF LIRE THEN & LIRE (P,S,BUF)
190 IF ECRIRE THEN & ECRIRE (P,S,BUF)
200 GOTO 120
```

Un programme de copie piste à piste en Applesoft se montre sensiblement plus lent que le COPYA de la disquette MASTER qui appelle un module binaire. On peut cependant en faire un programme très rapide en ajoutant quelques lignes :

```
31 F=0 : X=0 : B2=38400
41 IF F=0 THEN F=1 : & LIRCARTÉ(B2)
131 & CARTE (P,S,B2 LET x) : IF NOT x THEN 120
```

Ces trois lignes réduisent la copie à celle des secteurs occupés, car la commande CARTE permet de sauter la partie lecture/écriture chaque fois que le secteur est libre et donc que x=0.

La copie de sauvegarde de fichiers en cours de remplissage privilégie une copie sélective.

Ce programme se trouve sur la disquette du Disk Manager sous le nom COPIE CARTE, mais on y trouve également la conversion en assembleur, qui en fait un programme extrêmement rapide.

COPIE FICHER PAR FICHER

La copie doit se faire par blocs de secteurs puisque la mémoire libre ne permet pas forcément de copier un programme entier en une seule passe. Elle se fera avec les étapes suivantes :

- charger les TSL du programme à copier
- charger autant de secteurs que la mémoire peut en contenir à partir des TSL

- vérifier que la disquette objet offre une place suffisante
- vérifier si le programme n'existe pas déjà. S'il existe et qu'il n'y pas assez de place, déduire la taille du programme présent
- chercher de la place pour la ou les TSL. Ecrire dans chacune l'adresse de la suivante
- ajouter ou remplacer le nom du programme au catalogue avec ECRNOM
- chercher de la place pour chaque secteur chargé, et le recopier en mettant la carte à jour avec INSCARTE
- recharger la suite du programme ou fichier

EMPLOI DES COMMANDES DU DISK MANAGER

Chargement

La commande LIRNOM nous donnera toutes les indications utiles sur le programme à copier : adresse de sa TSL, longueur. La TSL sera chargée par LIRE et on verra en position 1 et 2 s'il y a les coordonnées d'une autre TSL pour ce programme.

A partir de la position 12, on trouve par PEEK les couples de coordonnées de chaque secteur du programme, qui sont alors données aux paramètres à la commande LIRE.

La mémoire libre est paramétrée en fonction de PEEK(110) et PEEK(112), comme précédemment dans le programme de copie piste à piste.

Recopie

LIRNOM recherche le même programme, et obtient sa longueur. S'il existe déjà, on peut effacer ou renommer l'autre version.

La commande ESPACE, avec déduction éventuelle de la longueur, permet de vérifier - par simple comparaison avec la longueur du programme à copier - si l'espace est suffisant.

Le chargement de la carte s'impose pour guider la recherche d'espace libre de la disquette cible. Un premier secteur sera affecté à la TSL de copie, celle-ci étant complétée avec l'adresse des secteurs et de la TSL suivante éventuelle. Dans le cas où le programme comporte plus de 122 secteurs, les opérations précédentes sont répétées.

Pour finir, ECRNOM ajoutera le nom et les coordonnées de la première TSL dans le catalogue.

VIII - Programmes

VIII-1 Ultra-copie

Encore un programme de copie ! Pourtant, nombreux sont ceux qui utilisent des fichiers de façon constante.

C'est toujours à titre d'application du Disk Manager que ce programme a été élaboré, et il démontre l'intérêt de pouvoir écrire un programme faisant appel au disque directement à partir du Basic (langage algorithmique) avant conversion en Assembleur (langage d'exécution).

Cette conversion est en fait nécessaire seulement parce que le programme comporte l'opération répétitive de lecture et écriture de 560 secteurs !

ULTRA-COPIE est sauvegardé sur la disquette jointe comme un programme Applesoft. Celui-ci débute par l'instruction CALL 2061 qui appelle le programme assembleur englobé par les pointeurs de l'Applesoft (et qui se situe après cette ligne en Basic).

ULTRA-COPIE est un utilitaire de backup auto-sélectif.

Afin d'augmenter la vitesse de copie, une option permet de ne prendre en compte que les secteurs occupés sur la disquette source. Cette option est prise automatiquement par le programme après calcul du nombre de secteurs occupés et estimation d'un gain en nombre de remplissage mémoire.

La touche 6 permet de commander impérativement la copie de tous les secteurs ou la copie sélective. Cinq autres touches fonctionnent selon le même principe.

La liste des touches et des fonctions concernées est la suivante :

- 1 : changement de slot source
- 2 : changement de slot objet
- 3 : changement de drive source
- 4 : changement de drive objet
- 5 : configuration mémoire
- 6 : force le mode de copie en intégral ou sélectif

VIII-2 UTILI-DISQUE

Les DOS standards tels que CP/M ou MS-DOS sont accompagnés de nombreux utilitaires, dont la diversité va bien au-delà de ce que propose le DOS 3.3 d'Apple. Pour compléter celui-ci, dans l'esprit du "support extérieur" volontairement favorisé par Apple et cher à ses fondateurs, le club CALL-APPLE a diffusé une disquette comportant une série d'utilitaires réalisés par les étudiants du Lawrence Hall of

Science, qui a eu beaucoup de succès.

UTILI-DISQUE comprend dans un même programme Applesoft la plupart de ces utilitaires qui avaient été écrits en Assembleur. Il possède en outre diverses commandes qui sont disponibles sur les DOS standards, ainsi que des commandes inédites.

Quelle que soit la commande considérée, il n'effectue aucune opération qui laisse à l'utilisateur la responsabilité de décider si elle peut être effectuée sans danger : toute commande qui modifie une disquette effectue d'abord un contrôle qui prévient contre le danger d'effacement de programmes.

En outre, une commande modifiant l'organisation de la disquette ne peut être interrompue en cours d'opération par CTRL-C.

Des utilitaires généraux de gestion du disque effectuant certaines opérations d'UTILI-DISQUE ont déjà été diffusés et ont reçu les louanges des reviewers. La réunion sur une même disquette de trois utilitaires disque à succès démontre l'avance et les possibilités du Disk Manager !

Options de UTILI-DISQUE

<A>ctiver
ackup
<C>orriger la carte
<D>elete
<E>nvironnement
<F>ix
<L>ecture d'adresses
<O>ccupation
<R>écupération d'espace
<S>canner
<T>ri alphabétique
<U>ndelete
<V>érifier

Activer

Connecte le premier ou le second drive, spécifié par l'option environnement.

Backup

- Dos : copie du DOS d'une disquette sur une autre, vérifie que les pistes où il doit être placé n'ont pas déjà été réutilisées.
- ULTRA-COPIE : il peut être chargé à partir de UTILI-DISQUE.

Corriger la carte

Cette option reconstruit la carte d'occupation à partir du catalogue, mais avec des paramètres fixes : le catalogue est lu sur la piste 17. Le DOS est considéré présent si aucun programme ne réutilise les pistes 1 ou 2. On peut modifier tous les paramètres sur le listing qui occupe les numéros de ligne de 3000 à 3999.

Delete

Effacement de programmes en série. Répondre "=" pour effacer tout le catalogue.

Environnement

Met à jour la configuration disponible.

Fix

Vérifie et corrige la taille des programmes. La carte d'occupation est en outre corrigée pour tenir compte de tous les secteurs occupés par les programmes.

Lecture adresses

Donne les adresses d'un programme binaire ou de tous ceux que la disquette contient, selon la commande passée.

Occupation

Affiche sur l'écran l'occupation de la disquette, avec un espace pour un secteur libre et un X pour un secteur occupé. Les secteurs occupés normalement par le DOS et le DIRECTORY sont en inverse.

Quitte

Retour au Basic.

Récupération

Deux options :

- supprimer le DOS : récupère les pistes 1 et 2 pour les programmes et fichiers, mais en vérifiant qu'elles n'ont pas déjà été libérées et utilisées.

- récupérer une partie du DIRECTORY : quand on stocke peu de programmes sur une disquette, on peut au besoin récupérer les secteurs 1 à 14 du catalogue.

Le programme U-D de la disquette Disk Manager vérifie que l'on n'efface pas une partie utile du catalogue, en tenant compte des programmes ou fichiers antérieurement DELETés. Il vérifie également que l'on n'a pas déjà utilisé ces secteurs.

Scanner

Vérifie l'absence de mauvais secteurs sur une disquette initialisée, sans effacer ce qui s'y trouve. Les mauvais secteurs sont effacés de la carte et, quand ils appartiennent à un programme, son nom est affiché; on demande alors s'il faut le supprimer.

Tri alphabétique

Classement alphabétique du catalogue, et purge des programmes DELETés. Lorsque la procédure de réécriture du catalogue est commencée, elle ne peut plus être interrompue par CTRL-C, qui alors pour effet de la faire reprendre au début.

Undelete

Un programme DELETé peut être restauré si aucun de ses secteurs n'a été réutilisé par un autre programme. C'est ce que vérifie cette option avant de replacer le programme sur le catalogue.

Vérifier

Correspond à la commande VERIFY, sur un programme ou en série.

IX - EDICAT

Le programme d'application d'édition de catalogue n'opère pas directement sur le disque, mais reconstitue en mémoire le catalogue (directory) et affiche une version simulée de celui-ci. C'est sur cette version que sont effectuées les commandes : déplacement de noms de programmes, création de titres, suppression, changement de noms, verrouillage et déverrouillage.

Seule l'instruction DELETE est réalisée simultanément sur le disque et en mémoire; l'option UNDELETE permet alors de réparer une fausse manœuvre éventuelle.

DELETE agit également sur la carte d'occupation de la disquette; c'est pour cela que l'on effectue cette opération simultanément sur la disquette.

Le menu des commandes apparaît sur la dernière ligne de l'écran. Cette ligne joue également le rôle du curseur, et le menu s'efface pour faire apparaître le programme actuellement sélectionné, quand une des commandes a été choisie.

Le programme EDICAT fonctionne en mode interactif, chaque ordre ayant un effet immédiat sur la liste des programmes affichée.

Placer la disquette dont le catalogue doit être édité dans le drive, indiquer le numéro de slot et de drive, ou appuyer sur RETURN pour conserver par défaut les valeurs affichées. En indiquant E comme numéro de slot, on retourne au catalogue en cours d'édition.

La disquette sur laquelle on lit le catalogue à éditer ne doit pas être retirée jusqu'à l'édition d'un autre catalogue. Certaines opérations s'effectuent en effet simultanément en mémoire et sur le disque. Les opérations s'enchaînent en série à l'aide d'un curseur : on utilise la touche "<-" pour remonter et "->" ou toute autre touche pour descendre.

Une fois l'édition du catalogue terminée, deux options sont possibles :

- 1 - réécriture du catalogue à partir de la liste en mémoire
- 2 - retour au menu sans réécriture

Le menu affiche la liste des commandes interactives :

LISTE DES COMMANDES

- A = classement alphabétique des programmes
C = simulation en mémoire du catalogue tel qu'il apparaîtra à la commande "CATALOG"
D = destruction d'un programme. Si le nom effacé appartenait au catalogue chargé, il y a simultanément destruction sur le disque
F = insertion d'un titre sous forme de nom de fichier. Il peut être en inverse et affiché à partir de première colonne de l'écran lors du "CATALOG". Le texte est auto-centré au milieu d'une zone de 30 caractères (23 avec l'option colonne 1) et complété par des espaces.
I = affichage des commandes
L = verrouillage
N = RENAME
P = purge des noms des programmes DELETés
Q = fin d'édition
R = restauration d'un programme DELETé. Cette restauration se fait simultanément sur le disque si le programme y figurait.
T = transfert d'un nom à un autre endroit du catalogue. On appuie sur RETURN quand le curseur est placé sur le nom à transférer; on le place sur le nom du programme devant lequel il doit apparaître et on fait <RETURN>.
U = déverrouillage
<-,> = déplacement du curseur
RETURN = lance l'opération voulue
ESC = abandonne la procédure sélectionnée

Le curseur

Il affiche en bas de l'écran le nom du programme sur lequel il est placé. On ne peut pas le placer devant tous les noms : une sélection logique s'opère, un programme DELETé, ne pouvant être DELETé de nouveau, sera sauté par le curseur. Par contre, aucune sélection n'est faite pour les commandes de verrouillage et déverrouillage.

X - MULTI DISQUE

Combien de disquettes possédez-vous ? Il serait intéressant de faire la liste des programmes qui y figurent, pour avoir sur papier la localisation de chacun d'entre eux, notamment si cette liste peut être triée alphabétiquement ! Les informations données par le listing de MULTI DISQUE sont :

- le numéro de la disquette
- le type des programmes

- la taille des programmes

Rien n'interdit d'ajouter d'autres informations telles que les adresses des programmes binaires ou les adresses des TSL, le programme étant écrit en Basic et facile à modifier.

Les possibilités suivantes figurent dans le programme MULTI DISQUE de la disquette Disk Manager :

A : ajouter un catalogue à la liste en mémoire. Placer la disquette et donner un numéro de volume, puis appuyer sur RETURN.

L : affichage sélectif sur numéro de volume, type de programme, nom entier ou premières lettres de noms, ou sur un ensemble de ces critères... Les numéros de volume comportent de 1 à 3 chiffres. En indiquant "6-" comme numéro de volume, on obtient les numéros en 60, avec "61-" les numéros en 610.

Barre d'espace : affichage du listing complet

C : classement alphabétique ou par volume

E : effacement du fichier en mémoire

F : affichage des fonctions non présentes au menu

I : impression de la liste. Avec les options de listing sélectif ou en entier, en appuyant sur RETURN à chaque option. Indiquer le numéro de slot.

M : nettoyage de la mémoire et affichage de l'espace libre

R : chargement d'un fichier

S : suppression de volume

W : sauvegarde d'un fichier

/ : catalogue

La capacité par fichier est de 1000 programmes (avec 48K).

Tri en assembleur

Il s'agit d'une routine de tri de deux tableaux en assembleur; le premier tableau est celui des noms, le second un index. Cette routine peut avantageusement remplacer la routine en Applesoft, assez lente sur de long listings bien que l'algorithme de tri dichotomique choisi et le classement sur index soient des plus performants.

Le programme définit 5 tableaux :

DB\$() : noms de programmes

I() : index

V() : numéro de volume

T\$() : type

L\$() : longueur

Le nom et les paramètres affichés en position n sur le listing n'occupent pas la position n dans ces tableaux; seul l'indice I() correspond à sa place physique et contient la position de la place physique des autres éléments dans les tableaux.

Le tri donc peut s'opérer avec les deux tableaux DBS() et I() à condition de faire correspondre les places physiques des noms et des indices, le classement ayant besoin au moins de deux tableaux.

Il faut alors modifier le restant du programme afin que les noms et indices correspondent à leur places physique et les autres paramètres à la position contenue dans I(); cela concerne les routines :

- d'affichage (2000-)
- de sauvegarde (6000-)
- de chargement (7000-)
- de suppression de volume (5000-)

Suppression de volume

Elle s'opère formellement en mémoire par suppression dans l'index, mais non physiquement par compression du fichier. Celle-ci est réalisée lors de la sauvegarde; la sauvegarde suivie d'un nouveau chargement peut donc s'avérer utile pour récupérer de l'espace dans un fichier plein.

Fichier

La base de données est stockée sur un fichier séquentiel ordinaire. Ce fichier contient les noms, les numéros de volume, de type et les longueurs. Lors du stockage de la base, chaque élément retrouve la même place physique que son indice de classement. Ceci dispense donc de stocker les indices, qui sont établis a priori. Cela permet également de recharger un fichier sur un programme modifié, notamment avec une routine de tri assembleur.

A1 : prévenir les problèmes

Problèmes possibles en mémoire :

- la variable d'adresse de buffer n'ayant pas été initialisée, elle contient la valeur zéro, et le chargement d'un secteur se fait à cette adresse.
- avec les commandes LIRADR, on entre dans une boucle sans fin si on ne teste pas la fin du catalogue sur la variable de nom ou numériquement.

Problèmes possibles sur disque :

- lors de la mise à jour de la carte d'occupation, ne pas omettre quand on commande une instruction de nom (LIRNOM ..) de spécifier un buffer par AFFECTER, sans quoi le buffer en cours s'efface.
- changement de drive : se rappeler que, si l'on change de drive, toute commande de carte recharge la carte de la nouvelle disquette sur le buffer en cours.

A2 : adresses du DOS et utilisation

Ces adresses supposent une version du DOS 3.3 non modifiée. Sont fournies ici quelques adresses susceptibles d'être utilisées par un programme Basic ou assembleur, ou même en mode direct. On pourra disposer d'une liste plus détaillée dans "Beneath Apple Dos" de Don Worth et Peter Lechner ou "DOS Source" désassemblé par Randy Hide.

\$76 Vaut \$FF (255) en mode direct; toute autre valeur signifie qu'un programme est en cours. On peut passer des commandes comme INPUT ou GET, normalement interdites en mode direct, avec POKE 118,0.

\$3D0 Reconnecte le DOS sans modifier de paramètre présent

\$3D3 Reconnecte le DOS comme s'il venait d'être chargé et efface le programme Basic en mémoire

\$3D6 Pointe sur la routine de gestion de programmes et fichiers

\$3D9 Pointe sur la routine de lecture/écriture ou RWTS

\$3DC Pointe sur la liste des paramètres utiles à la routine de gestion de fichiers. S'appelle immédiatement avant \$3D6.

\$3E3 Pointe sur l'IOB qui contient tous les paramètres utiles à la routine RWTS et s'appelle immédiatement avant \$3D9

\$3EA Reconnecte le DOS et les pointeurs d'entrée clavier et affichage. Doit être appelé quand on réaffecte l'adresse pointée par RESET. Exemple :

\$3F2 : 00

\$3F3 : 03

\$300 JSR \$3EA
JSR \$D566
JMP \$D742

Ce petit programme redémarre l'exécution d'un programme Basic quand on appuie sur RESET.

Doit être appelé également après que l'on ait modifié les pointeurs d'entrée ou sortie de caractères (KSL/CSL)...

\$3F2-\$3F3 Adresse pointée pour branchement lors d'un RESET

\$3F4 Complément à \$A5 de l'adresse contenue dans \$3F3. Si la valeur ne donne pas le complément, appuyer sur RESET provoque un boot. Après avoir modifié cette adresse, un appel à \$FB6F (-1169) calcule et stocke le complément.

\$9600 Début du buffer 3

\$9D00 Début du DOS. Contient l'adresse du premier buffer. Pour placer un programme binaire entre le DOS et ses buffers, soustraire à l'adresse contenue en \$9D00 la longueur du programme augmentée de 38, puis appeler \$A7D4 pour que le DOS reconstruise ses buffers.

\$9D1E-9D55 Adresses des commandes DOS

\$9E42 Type du programme à charger lorsque le DOS est booté : placer \$34 pour un programme binaire ou \$14 pour un fichier EXEC. Directement sur la disquette, c'est en position \$C du premier secteur, à la place de \$42.

\$A021 Interdit l'usage en mode direct des commandes OPEN, READ etc... Placer \$AD (173) à cette adresse (40993) pour passer outre.

\$A1AE Met à 0 la liste des paramètres de gestion de fichiers

\$A1D6 Routine de conversion décimale

\$A203 Routine de conversion hexadécimale

\$A316 Fermeture de tous les fichiers

\$A54F-6D INIT

\$A56E Commande CATALOG selon les paramètres drive et slot passés dans la dernière commande

\$A57A FP

\$A59E INT

\$A6D5 Prise en charge d'erreur. Branche en \$A6EF si ONERR .. GOTO actif, sinon affiche le message d'erreur.

\$A6EF Retour au Basic ou branchement sur la routine d'erreur Basic si

un programme est en cours

702 Affiche le message d'erreur dont le numéro est chargé dans le registre X

\$A884-A908 Table des textes de commandes du DOS

\$A941-A94A Table des abréviations des commandes

\$A955-A970 Table des valeurs limites acceptées pour chaque commande :

Volume : \$A955-56 limite mini 00 00

\$A957-58 limite maxi FE 00

Drive : \$A959-5A mini 01 00

\$A95B-5C maxi 02 00

Slot : \$A95D-5E mini 01 00

\$A95F-60 maxi 07 00

Longueur : \$A961-62 mini 00 00

\$A963-64 maxi FF 7F

Numéro d'enregistrement : \$A965-66 mini 00 00

\$A967-68 maxi FF 7F

Byte : \$A969-6A mini 00 00

\$A96B-6C maxi FF 7F

Départ de programme binaire : \$A96D-6E mini 00 00

\$A96F-70 maxi FF FF

La valeur FF 7F correspond à 32767, et FF FF à 65535.

Le DOS accepte le stockage de programmes binaires de longueur supérieure à 128 secteurs, si on remplace la valeur \$7F en \$A964 par une valeur supérieure.

\$A971-AA3E Table des textes des messages d'erreur

\$AA57 (43607) MAXFILES par défaut, normalement 3

\$AA60-61 Longueur du dernier programme binaire chargé

\$AA72-73 Adresse de départ du dernier programme binaire chargé

\$AA75-92 (43637-666) Buffer du nom de programme. Le nom du programme HELLO est placé ici avant l'exécution de la commande INIT, ainsi que le nom du programme concerné lors de toute commande de gestion disque.

\$AA93-B0 Second buffer de nom de programme, sert à la commande RENAME

\$AAB1 Contient le nombre de buffers par défaut : 3

\$AAB2 Contient CTRL-D, soit \$84

\$AAB3 Indicateur de commande EXEC en cours : a une valeur différente de 0 quand EXEC est actif. Placer à cette adresse 0 par POKE (43699), dans un fichier EXEC, pour y intégrer des commandes INPUT ou GET; placer également zéro en 118, restituer 255 en 118 et 1 en 43699 avant de poursuivre la procédure EXEC.

\$AAB6 Code indiquant un programme en cours : \$40 si Applesoft, 0 si Integer

\$AAB8 Texte de "APPLESOFT"

\$AAC1-C2 Adresse du RWTS (\$B7E8)
 \$AAC3-C4 Adresse du buffer VTOC (\$B3BB)
 \$AAC5-C6 Adresse du buffer directory (\$B4BB)

\$AE42 Affiche la valeur placée en \$44 sur trois digits, utilisée pour donner le numéro de volume

\$ABCD Initialisation de la zone de gestion de fichiers

\$ADA9-B5 Affiche "DISK VOLUME" puis poursuit la procédure CATALOG

\$ADB6-BD Affiche le numéro de volume puis poursuit la procédure CATALOG

\$ADD9 Test qui exclue les programmes DELETés lors de l'affichage du catalogue : placer \$10 (16) à cette adresse (44505) pour afficher les programmes DELETés seuls. Placer 1 pour afficher tous les programmes, \$30 (48) pour retrouver l'affichage normal.

\$AE17 (44567) Largeur des programmes affichés lors du CATALOG

\$AE22 Formatage du catalogue : placer \$AD (173) à cette adresse (44578) et 12 en 44567 pour obtenir un catalogue sur 2 colonnes

\$AE8E-\$AF07 Fonction INIT

\$B3A7-AE Initiales des types de fichiers reconnus par le DOS :
 T,I,A,B,S,R,A,B.
 Ils correspondent aux codes : 00, 01, 02, 04, 08, \$10, \$20 et \$40

\$B3AF-BA Texte "DISK VOLUME " stocké en sens inverse

\$B3BB-B4BA Buffer réservé au chargement de la carte d'occupation de la disquette

\$B3F3 Début de l'allocation espace dans la carte d'occupation chargée

\$B4BB-B5BA Buffer réservé au chargement des secteurs du directory

\$B5BB-DO Liste des paramètres de gestion de fichiers

\$B5D1-FD Zone de travail de la gestion de fichiers

\$B6B3 Zone utilisée par les dernières modifications du DOS 3.3, libre auparavant.

\$B74A-8C Place le DOS sur les pistes 0, 1 et 2 en fin l'initialisation. Peut être appelé directement si certains paramètres sont définis avant appel :

\$00 en \$AA5F marque la fin de INIT
 \$00 en \$B7B0 ! pointe sur le
 \$9D en \$B7F1 ! début du DOS
 \$00 en \$B7EB volume universel

\$B7E8-F8 Input/Output Block (IOB) : c'est le bloc d'entrée/sortie contenant les paramètres nécessaires à tout accès disque par RWTS.

\$B7E8 : numéro de table (inutile)

\$B7E9 : slot * 16 normalement \$60

\$B7EA : drive, 1 ou 2

\$B7EB : volume ou 0

\$B7EC : piste à lire

\$B7ED : secteur à lire

\$B7EE : ! adresse

\$B7EF : ! DCT

\$B7F0 : ! adresse du buf-

\$B7F1 : ! fer de données

\$B7F2 : inutilisé

\$B7F3 : inutilisé

\$B7F4 : code commande : 01=lecture

02=écriture

03=init

00=passage

\$B7F5 : code erreur ou dernier octet du secteur chargé

\$B7F6 : volume trouvé après opération

\$B7F7 : slot trouvé

\$B7F8 : drive trouvé

\$B7F9-FA Inutilisés

\$B7FB-FE DCT : Device Characteristic Table

\$B800-29 Convertit 256 octets de 8 bits en 342 nibbles de 6 bits

\$B8C2-DB Convertit 342 nibbles de 6 bits en 256 octets

\$BA69-95 Zone libre sur Apple II, utilisée sur Apple //e.

\$BB00-FF Premier buffer

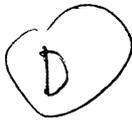
\$BC00-55 Second buffer

\$BD00-18 RWTS : est appelé avec l'adresse de l'IOB en Y (bas) et A (haut) ou par JSR \$3E3 immédiatement suivi de JSR \$3D9.

\$BEFE (38894) Nombre de pistes normalement initialisées, soit \$22. En plaçant \$23 (35), on fait initialiser une piste supplémentaire.

A3 : carte de la mémoire centrale

! Adresse en !	! Adresse en !	!	!
! décimal !	! hexadécimal !	!	! Utilisation !
! 40192	! \$9D00	!	! Début du DOS !
! 39590	! \$9AA6	!	! Buffer unique !
!	!	!	! avec Maxfiles1 !
! 39168	! \$9900	!	! Buffer DM 4 !
! 38912	! \$9800	!	! " 3 !
! 38656	! \$9700	!	! " 2 !
! 38400	! \$9600	!	! " 1 ou !
!	!	!	! buffer DOS a- !
!	!	!	! vec Maxfiles 3 !
! 36608	! \$8F00	!	! D.O.M !
! 34048	! \$8500	!	! Disk Editor !
! 33792	! \$8400	!	! Dernière page !
! >	! >	!	! Mémoire libre !
! >	! >	!	! id. !
! >	! >	!	! id. !
! 2048	! \$800	!	! id. !
! 1024	! \$400	!	! Mémoire écran !
! 976-1023	! \$3D0-3FF	!	! Pointeurs !
! 768-975	! \$300-3CF	!	! Mémoire libre !
! 24-31	! \$18-1F	!	! Paramètres du !
!	!	!	! Disk Manager !



Avertissement

Les manipulations avec les noms de fichiers, comme le tri, supposent que vous n'avez pas de fichier qui, comme c'est le cas sur la disquette du Disk Manager, comporte des noms avec des caractères en inverse ou en clignotant. De toute façon, méfiez-vous de ces noms de fichiers qui provoquent des problèmes quand on tente d'obtenir un simple catalogue sur disquette.

Erratum

Page 21, troisième ligne avant la fin : la dernière commande s'obtient avec le point ".", pas le point d'exclamation "!".

Page 34, lire pour la ligne 140 :

```
140 IF (P>FIN) AND LIRE THEN BUF = PLAFOND - BUF
```

Page 41 : l'option "M" est supprimée, car elle provoquait de dangereuses "collisions" en mémoire.

Pages 41 et 42 : les explications "Tri en assembleur" et "Fichier" sont à présent inutiles, car ces modifications ont été apportées dans la présente version du Disk Manager.

