

JASMIN

T.R.O.O.® (S.A.R.L.)



TECHNOLOGIE RECHERCHE & APPLICATIONS NOUVELLES

53, Impasse Blériot
Parc des Savels

83130 LA GARDE
Tél : (94) 21.19.68

EDITION AOUT 1984

NOTE IMPORTANTE POUR LE BRANCHEMENT DU "JASMIN"

Pour la version double boîtiers, sur l'arrière du boîtier contrôleur, il existe cinq câbles

- 1 câble alimentation 220 VOLTS
- 1 câble plat de 34 brins à connecter sur le BUS d'extension de ORICI ou ATMOS
- 1 câble plat de 34 brins avec 2 connecteurs encartables prévus pour brancher sur 2 lecteurs.
- 2 câbles alimentation 5 Volts et 12 Volts avec 2 connecteurs (à 4 broches mais seulement 3 fils) munis d'un détrompeur, à brancher sur 2 Lecteurs.

Pour le cas où il n'existe qu'un seul lecteur, un connecteur 34 broches et un connecteur 4 broches restent réservés pour une future extension de 2ème lecteur.

Les connecteurs 34 brins de câble plat destiné aux lecteurs n'ont pas de détrompeur : une marque blanche est inscrite sur le connecteur et doit impérativement se trouver sur la partie supérieure, lors du branchement, sous peine de graves dommages.

Connecter le connecteur 34 brins, point blanc vers le haut, et celui de l'alimentation 5 volts et 12 Volts sur le lecteur (petit détrompeur en bas)

PREFACE

Nous tenons à remercier tous nos amis qui ont eu raison de croire dès le début au grand succès du JASMIN et qui y ont participé avec passion depuis sa conception jusqu'à sa fabrication. C'est grâce à la variété de leurs professions et leurs remarques d'utilisation que nous obtenons maintenant un appareil à la fois puissant, pratique et simple d'utilisation.

La mise au point finale a été longue et laborieuse. Nous avons rencontré d'énormes difficultés pour rendre le complexe système d'exploitation T.DOS compatible entre l'ORIC et l'ATMOS. Vous remarquerez d'ailleurs pour certaines instructions, une variation dans la syntaxe.

La version V2 - 1 pour ORIC 1 et V2 - 2 pour ATMOS comporte plus de 40 instructions. Nous espérons ainsi satisfaire les possesseurs des anciennes versions et nous excuser du retard pour la diffusion de celle-ci. Comme tout système complexe d'exploitation, il subsiste toujours des erreurs qui ne se remarqueront qu'à l'usage. Nous vous demandons de bien vouloir nous faire part de vos remarques en cas de détection de défauts. Cela nous permettra de mettre à jour le T.DOS, voire de l'améliorer au fur et à mesure. Vous pouvez d'ailleurs obtenir chez le revendeur le plus proche, la mise à jour gratuite de votre T.DOS.

Pour vous permettre de profiter dès à présent de cette nouvelle version du T.DOS, nous avons fait éditer d'urgence ce livret. Un manuel plus complet qui vous permettra de tirer le maximum de profit du T.DOS est en cours de rédaction et sera diffusé en librairie. N'hésitez pas à nous contacter pour tous renseignements.

TRAN THO NGUYEN

SOMMAIRE

CHAPITRE I INTRODUCTION

A- Démarrage et Première utilisation	2
--	---

B - NOTION DE FICHIER

B-1 notions de Fichiers sur deux exemples	6
B-2 nécessité d'autres types de Fichiers	9

C - NOTION DE DISQUETTES

C-1 disquette matérielle	10
C-2 disquette logique	12
C-3 numéro logique de Fichiers	13

CHAPITRE II UTILISATION DU DOS

A - OPERATION SUR LES DISQUETTES

A-1 système mono disquette et multidisquette	15
A-2 formatage	15
A-3 initialisation	16
A-4 copie de T.DOS	16
A-5 nomination	17
A-6 catalogue sur disque	17
A-7 liste du catalogue	18
A-8 copie entière de disquette	18
A-9 montage d'un volume	18
A-10 démontage d'un volume	19
A-11 copie de Fichier	19
A-12 copie de Fichier Binaire pour un mono-lecteur	20

B - TRAITEMENT SUR TOUT FICHIER

B-1 sauvegarde	! SAVE	21
B-2 chargement	! LOAD	22
B-3 chargement et exécution	! "PROGRAMME" ..	22
B-4 transfère un fichier de cassette à disquette sans déprotéger	! "TKD"	23
B-5 recherche de Fichier	! SEARCH	24
B-6 changement de nom	! RENAME	25
B-7 protection d'écriture	! LOCK	25
B-8 déprotection d'écriture	! UNLOCK	25
B-9 concaténation d'un Fichier Basic à la suite d'un autre Fichier Basic	! MERGE	26
B-10 destruction de Fichier	! DEL	26
B-11 effacement de queue	! CUT	26

C - TRAITEMENT D'ERREUR

C-1 empêche l'interruption en cas d'erreur d'utilisation du DOS	! ERSET	27
C-2 autorise l'interruption normale en cas d'erreur d'utilisation du DOS	! EROFF	27
C-3 en cas d'erreur d'utilisation du DOS, continue à la ligne n°	! ERR GOTO	28

D - FICHIERS DE DONNEE DE TYPE :.DAT

D-1 généralités	29	
D-2 protection d'écriture	! WL	29
D-3 déprotection d'écriture	! WUL	31
D-4 ouverture de Fichier	! OPEN	32
D-5 fermeture de Fichier	! CLOSE	32

E - FICHIER A ACCES SEQUENTIEL

E-1 création d'un fichier	! CREATE	32
E-2 ouverture de Fichier	! OPEN	33
E-3 fermeture de Fichier	! CLOSE	33
E-4 écriture	! WRITE	33
E-5 lecture	! TAKE	34
E-6 positionnement début de Fichier	! REWIND	35

E-7	positionnement fin de Fichier	! APND	35
E-8	indication du n° de rubrique courante du fichier et du nombre total de rubriques	! WHERE.....	36
E-9	incrémentation du pointeur du Fichier	! JUMP	36

F - FICHER A ACCES DIRECT

F-1	création de Fichier	! CREATE ...	37
F-2	écriture	! WRITE ...	37
F-3	lecture	! TAKE ...	38
F-4	Indication du nombre total d'enregistrement ...	!WHERE	39

G - OPERATIONS SUR LES MATRICES

G-1	sauvegarde de la Matrice ou du Tableau dans un Fichier	! MSAVE	39
G-2	chargement du Fichier dans la Matrice ou le Tableau	! MLOAD	39

H - SAUVEGARDE D'ECRAN

H-1	en basse résolution	! LSCR.....	40
H-2	en haute résolution	! HSCR	40

I - EXECUTION AU DEMARRAGE

I-1	assignation pour le démarrage	! START ..	41
I-2	pas de démarrage automatique	! UNSTART ..	41

CHAPITRE III

ANNEXE

A-1	occupation Mémoire	42
A-2	comment entrer dans la MEV parallèle	42
A-3	adresses du Contrôleur	43
A-4	table des messages d'erreur	43
A-5	exemples	44
A-6	garantie	45

CHAPITRE I - INTRODUCTION

DESCRIPTION SOMMAIRE - MISE EN ROUTE

NOTE TRES IMPORTANTE

Le Lecteur de Disque JASMIN est un appareil à haute précision et qui demande certaines précautions d'emploi :

- les chocs lui sont évidemment préjudiciables.
- il ne faut jamais, au repos et surtout en cas de déplacement ou transport de l'unité de Disquette JASMIN laisser une disquette dans son logement, car dans ce cas les têtes de lecture sont en position de fonctionnement et l'appareil est susceptible d'être endommagé. Dans ce cas la garantie ne sera pas appliquée.
- Les lecteurs utilisent des têtes très sensibles aux rayonnements électromagnétique, et par sécurité ne mettez pas votre lecteur à moins de 30 centimètres de votre Téléviseur ou Moniteur, ou encore d'autres appareils à rayonnement électro-magnétiques importants.
- En cas d'utilisation intensive, il est recommandé de diriger un ventilateur sur l'arrière de l'appareil, pour que les filets d'air refroidissent le lecteur.

A - Démarrage

En version mono-boîtier, de la gauche vers la droite, nous trouvons le lecteur ° 2, puis le lecteur ° 1, ensuite deux boutons situés à l'extérieur et à l'intérieur du boîtier et accessibles par la pointe d'un stylo à bille par exemple. L'un est le bouton R ou RESET de réinitialisation du système DISQUE, l'autre B ou BOOT fait démarrer ou redémarrer la disquette par appel du BOOTSTRAP.

En version double boîtier, un boîtier contient l'alimentation et la carte contrôleur, et chaque boîtier supplémentaire comprend une unité de disquette, reliée par câble à l'alimentation et à l'unité centrale.

PREMIERE UTILISATION

Pour vous permettre une première utilisation rapide, nous vous indiquons dès à présent la mise en route du système, avec les premières commandes pratiques. Plus loin, vous prendrez connaissance du détail des différentes instructions du T.DOS et de divers utilitaires.

L'ORIC 1 ou ATMOS ne doit pas être branché sur le JASMIN.

Brancher l'alimentation 220 Volts puis mettre en marche le contrôleur par l'interrupteur arrière : la diode rouge sur la face avant doit s'allumer (pour la version double boîtier).

Si tout marche bien, éteindre le contrôleur, débrancher l'alimentation particulière de l'ORIC 1 ou ATMOS et brancher le câble plat 34 brins sur le BUS d'extension de l'ORIC 1 ou ATMOS.

- Brancher la prise de JASMIN au réseau électrique
- Mettre sous tension par l'interrupteur situé à l'arrière du JASMIN.

Appuyer sur le bouton RESET. Attendre le message habituel d'initialisation de l'ordinateur.

- Insérer la Disquette MASTER sur la face ORIC ou ATMOS, suivant votre version dans le lecteur n° 1.
- Appuyer sur le bouton BOOT : un message d'initialisation apparaît et qui indique que vous êtes sous T.DOS.

© 1984 TECHNOLOGIE RECHERCHE ET APPLICATIONS NOUVELLES

READY

Que s'est-il passé ?

Un petit programme appelé "BOOTSTRAPPING" résident dans le système JASMIN (ROM coupleur) s'est déroulé et a effectué la lecture d'un fichier Système sur la DISQUETTE MASTER T.DOS, puis a placé le résultat de cette lecture en OVERLAY dans la mémoire 16 K RAM de votre unité centrale ORIC 1 ou ATMOS, laissant ainsi disponible les 48 K utilisateur de votre ordinateur.

En cas d'insuccès appuyer à nouveau sur RESET et BOOT.

- Attendre que READY apparaisse à nouveau sur l'écran.

SAUVEGARDE DU SYSTEME INITIALISATION DE DISQUETTE MAITRE

Après apparition de READY, votre lecteur est prêt à fonctionner, mais il est préférable de faire une sauvegarde du T.DOS, une fausse manoeuvre pouvant "écraser" le T.DOS, il vous faudrait demander une nouvelle copie à votre vendeur.

- Appeler le programme de Formattage par :
! "FORMAT" RETURN (en abrégé RTN)

Attendre le message :

..... FORMATTAGE ? O/N

Répondre O

Au message suivant :

..... CHANGER DE DISQUETTE

Ejecter la Disquette originale en appuyant sur la barre située en-dessous de la disquette. Introduire la disquette vierge, face A vers le haut.

Répondre O

Après formatage de la 1^{ère} face, une nouvelle demande intervient, répondre O et changer la face A par la face B.

Pour les lecteurs à DOUBLE TÊTE DE LECTURE, il est inutile et d'ailleurs impossible de retourner la disquette, car une seule position est possible.

Lorsque vous aurez formaté toutes vos disquettes, vous passerez au programme suivant : la copie entière de la disquette par l'instruction !"BKP" voir page correspondante.

UTILISATION D'UN DEUXIÈME LECTEUR

Le lecteur JASMIN est programmé pour l'utilisation d'un ou de deux Lecteurs, par les commandes suivantes :

! MOUNT "2." Monter le 2^{ème} Lecteur.
! DEMOUNT "2." Démonter le 2^{ème} Lecteur.

CHARGEMENT D'UN PROGRAMME BASIC

- Brancher JASMIN, placer une Disquette MAÎTRE, mettre sous tension, appuyer sur RESET, puis sur BOOT.
- Vérifier les programmes présents par la commande : ! CAT "" si vous avez un seul lecteur.
- Entrer votre programme, et s'il est sur cassette, faire CLOAD "xxxx" avec ou sans, S.
- Et le sauvegarder par :
! SAVE "1.xxx" RTN ou
! SAVE "1.xxxx.BAS" RTN
si vous signalez qu'il s'agit d'un programme BASIC
- Vérification du chargement par
! CAT "".
- Dans le cas de programme sur cassette la commande !"TKD" doit être utilisée de préférence.

TRANSFERT DIRECT DE CASSETTE A DISQUETTE

L'utilitaire ! "TKD"

permet le transfert direct d'un programme ou Fichier logé sur cassette. Il sera transféré directement sur Disquette avec le nom que vous lui donnerez ou conserverez. C'est un utilitaire conversationnel, il suffit de faire ce qu'il vous demande.

APPEL D'UN PROGRAMME BASIC

- L'ordre de chargement d'un programme BASIC est très simple :
! "xxxx" RTN
Le programme se chargera seul et démarrera, comme AUTO sur Cassette.

APPEL DE LA TOUCHE HELP

En cas de difficultés, ou simplement pour connaître facilement le mode d'emploi du système, il vous suffit de faire appel à l'instruction suivante ! "HELP" RTN et l'Ecran affichera le menu des instructions T.DOS.

Les Programmes de TEST inscrits sur le CATALOGUE de la DISQUETTE vous permettent de bien connaître le fonctionnement et la Syntaxe des Fichiers. Servez-vous en et imprimez le LISTING.

Vous pouvez dès à présent vous exercer sur votre JASMIN et vous habituer à sa présence et ses facilités et surtout sa facilité. Ensuite, quand vous le désirerez, nous verrons plus loin le détail des instructions du système d'Exploitation du Disque du LECTEUR JASMIN, appelé T.DOS

B - Notion de Fichier

1- Notions de Fichiers sur deux exemples.

L'adjonction à un système micro-informatique, d'une mémoire de masse à accès rapide (relativement aux temps d'accès d'informations sur cassettes magnétiques) ouvre des horizons nouveaux à la programmation.

Sauvegarde aisée des programmes, stockage de données permanentes, augmentation virtuelle de la taille de la mémoire centrale, etc...

Cependant se contenter d'un **matériel** amène rapidement à une déconvenue par l'impossibilité d'une utilisation immédiate. La nécessité d'un **logiciel** est évidente. Celui-ci doit être d'un emploi simple sans être rigide et doit comporter une qualité importante : la fiabilité.

vous avez vu apparaître le terme de fichier (qualifié parfois de système).

Qu'est ce qu'un Fichier ?

Vous allez dire un fichier c'est un ensemble de fiches. C'est juste, cependant par extension, on donne le nom de fichier à tous ensembles d'informations cohérentes rassemblées dans une même entité.

Essayons maintenant de voir, par deux exemples, de quelle manière peuvent être utilisés des fichiers.

- 1 - 1 Premier exemple :

Vous avez une passion, disons la micro-informatique et vous êtes abonné à une revue mensuelle depuis déjà quelques années. Vous êtes donc en possession d'un petit tas de ces revues et vous vous êtes donc certainement trouvé devant le problème qui consiste à vous rappeler l'ensemble des articles traités. Vous vous dites à ce moment là qu'il serait pratique d'avoir un ensemble de fiches qui comporteraient chacune le sommaire d'une des revues. Vous avez besoin d'un fichier !

Comment le construire ?

1)- Tout d'abord il faut lui donner un nom (sur 8 caractères maximum). Le nom de la revue ou une partie de ce nom fera parfaitement l'affaire.

2)- Il faut décider du contenu et de l'ordre des "fiches". Pourquoi le guillemet ? tout simplement parce que maintenant on va abandonner le mot "fiche" pour parler d'**enregistrement**.

Quand vous regardez des fiches cartonnées et que vous les passez une à une, vous comptez 1,2,3, etc. Les enregistrements sont également numérotés de 1 à N.

Oui direz-vous, mais une fiche comporte en haut et à gauche le mois et l'année de parution. Pas de problème : votre première revue date de Mars 1978, elle constituera l'enregistrement numéro 1, celle de Avril 1978, le 2, etc... Certes, mais que faire pour la revue de Février 1979 que je n'ai plus ? Encore une fois aucun problème, l'enregistrement correspondant contiendra la rubrique "N'est plus en ma possession".

A partir d'un programme que vous écrirez, il vous sera donc facile de faire l'association entre le mois de Mars 1979 et le numéro d'enregistrement correspondant (13 pour cet exemple).

Vous voilà donc avec un **fichier** qui a un **nom** et dont chaque **enregistrement** à une signification.

Voyons maintenant le contenu de ces enregistrements. Tout simplement chaque enregistrement pourra être fait d'un texte qui reproduira le sommaire de la revue associée à cet enregistrement. Une précaution sera toutefois à prendre, il faut connaître le nombre maximum de caractères (blanc y compris) que peut faire le plus gros sommaire, pour déterminer la taille de l'enregistrement. C'est une évaluation qui est assez simple.

Cet exemple vous donne une utilisation possible (votre imagination en découvrira d'autres) d'un **fichier à accès direct** : la longueur des enregistrements est **fixe**.

Une remarque vient immédiatement : la longueur des sommaires est très variable, et par conséquent certains enregistrements ne sont pas utilisés au mieux (peu de caractères utilisés par rapport à la taille) aussi, il peut être souhaitable de s'affranchir du paramètre taille de l'enregistrement.

Ceci peut être fait en utilisant un **fichier à accès séquentiel**. Vous allez me dire, alors, pourquoi utiliser un fichier direct alors qu'un fichier séquentiel paraît plus simple ? La réponse exacte à cette question sera développée dans la partie plus théorique de ce

manuel. Une réponse simplifiée est que l'on accède plus rapidement à un enregistrement d'un fichier direct qu'à celui d'un fichier séquentiel.

Il vous faudra toujours chercher une optimisation à partir des deux formules suivantes :

GAIN DE PLACE = PERTE DE TEMPS DE TRAITEMENT

et sa négation

GAIN DE TEMPS DE TRAITEMENT = PERTE DE PLACE

Cela veut dire qu'à un moment donné dans votre programmation il vous faudra faire un choix entre le temps et l'encombrement.

Nous y reviendrons....

1 - 2 Deuxième exemple :

Où est donc passé mon bel argent ? Il arrive parfois que l'on se pose cette question à la fin d'un mois !.

Vous décidez donc à partir du mois prochain de comptabiliser toutes vos dépenses, afin d'une part d'en avoir la liste à la fin du mois et d'autre part d'en connaître le montant mensuel. Vous avez donc un fichier dont le nom est celui du mois (ouvrir un fichier c'est le créer s'il n'existe pas, et autoriser les accès - essayez l'analogie avec un compte bancaire) et chaque jour vous créez un enregistrement qui comporte un détail des achats effectués.

Un achat = un libellé + un prix
= une variable chaîne de caractères + une variable réelle.

Un jour = un enregistrement = un ensemble d'achats.

Chaque soir, vous effectuez cet enregistrement. Vient alors la fin du mois, vous voulez alors obtenir la liste des achats et le total des dépenses. Votre programme va alors lire un à un les enregistrements, (ce qui se transformera en jours), puis pour chaque enregistrement, éditer les achats et faire la somme des prix, ainsi que la somme générale puis l'éditer.

Vous voyez dans cet exemple que dans un premier cas on écrit les enregistrements les uns après les autres. C'est une exploration **séquentielle**. On est donc en présence d'un cas typique d'utilisation d'un fichier à accès séquentiel.

Pourquoi pas un fichier à accès direct : Parce que vous perdriez de la place sans gagner de temps puisque de toute manière il faudra écrire et surtout lire tous les enregistrements dans l'ordre, le but n'étant pas d'accéder aux dépenses d'un jour particulier mais de faire :

Une édition et une synthèse pour tous les jours du mois.

1 - 3 Enseignement à tirer des deux exemples :

Nous avons voulu montrer dans ses deux exemples la différence et l'utilisation rationnelle de deux types de fichiers à accès séquentiel.

Chaque type convient pour les deux exemples, mais le type direct convient mieux pour le premier exemple (simplicité et rapidité de recherche - la longueur de l'enregistrement pouvant être optimisée avec un peu d'astuce), et le type séquentiel au second (puisque l'on ne peut pas gagner de temps, autant gagner de la place !).

2 - nécessité d'autres types de Fichiers

Vous avez acquis maintenant quelques notions, et une idée sur deux types de fichiers :

- les directs
- les séquentiels.

Ce n'est pas suffisant, il est nécessaire d'avoir d'autres types de fichiers. Le premier type qui vous est venu à l'esprit est certainement le fichier BASIC.

Si vous êtes un programmeur chevronné pour qui l'assembleur 6502 n'a plus de secrets (des défauts mais pas de secrets), et même dans le cas contraire sachez tout de même que vos programmes assembleurs présents ou futurs pourront être sauvegardés en type BINAIRE.

Il en est de même des zones mémoires qui vous sont précieuses (telles des images particulièrement réussies).

Quand au type SYSTEME, comme son nom l'indique il est réservé au système, c'est à dire au T.DOS.

Ces trois types sont des types de base. Cependant par commodité pour vous utilisateurs, il existe trois autres types qui sont :

- LE TEXTE
- LE COMMANDE

Du point de vue du T.DOS, ils ne servent à rien, par contre, pour vous-mêmes, il vous permettront de faire une distribution en fonction de l'utilisation que vous ferez de ces fichiers. (Ce ne sont que des mnémoniques pour votre usage personnel).

En résumé, nous avons les types :

ECRAN : .SCR
 BINAIRE : .BIN
 BASIC : .BAS
 SYSTEME : .SYS
 TEXTE : .TXT
 DATA : .DAT
 COMMANDE : .CMD
 MATRICE : .ARY

Quand vous aurez étudié les diverses commandes, ces différences vous sembleront limpides.

Cependant, sachez tout de même, que si vous demandez l'exécution d'un fichier TEXTE, le T.DOS ne sera pas d'accord car les seuls fichiers **exécutables** sont de types BASIC, BINAIRE, SYSTEME, CMD..

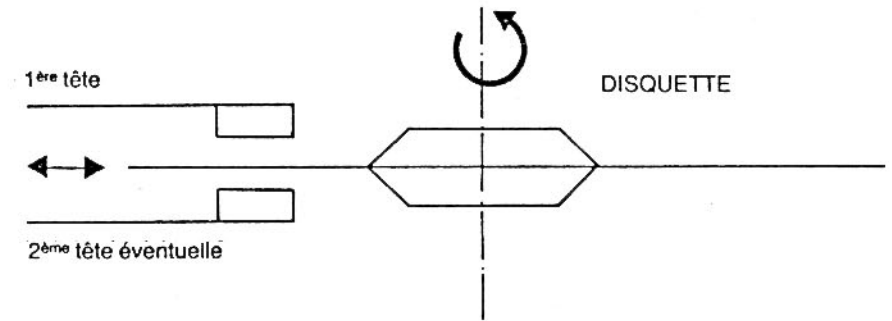
Tiens, voilà un nouveau terme : exécutable. Il signifie que le contenu du fichier peut être chargé en mémoire puis faire l'objet d'une commande RUN ou CALL..... (exécution d'un programme, commande éventuellement automatique).

C - Notions de Disquettes :

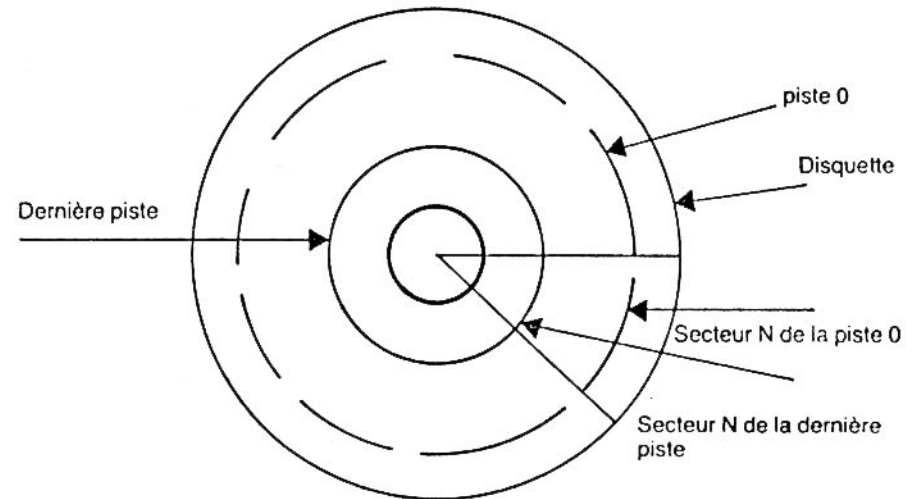
1 - Disquette matérielle :

Nous allons maintenant voir succinctement la structure d'une disquette. Comme son nom l'indique, une disquette est un petit disque recouvert d'un dépôt magnétique.

Cette disquette peut tourner autour de son axe central au-dessus et au dessous, pour un 2 TETES, se déplace une tête de lecture suivant le rayon de la disquette.



Le déplacement de la tête s'effectue cran par cran. Elle peut donc parcourir un certain nombre de cercles concentriques que l'on appelle des **pistes**.



Ces pistes sont découpées en secteurs (d'une capacité de 256 Octets).

Sur votre JASMIN vous disposez de :

- 41 pistes/face
- 17 secteurs par piste.

Ce qui fait 357 K Octets **formatés** :

Formatés : Encore un terme nouveau.

A l'origine une disquette neuve ne contient aucun enregistrement de quelque nature que ce soit.

Il est nécessaire de **formater** cette disquette, c'est à dire de créer le découpage des pistes en secteurs.

Une commande du T.DOS est prévue à cet effet.

ATTENTION :

Cette commande de formatage ne doit en aucun cas être utilisée sur la disquette qui supporte le système T.DOS. Cela conduirait à sa destruction.

2 - Disquette logique

Par Disquette logique, nous entendons une unité par laquelle vous n'avez pas à vous soucier de son fonctionnement ni même du lieu physique d'implantation des données.

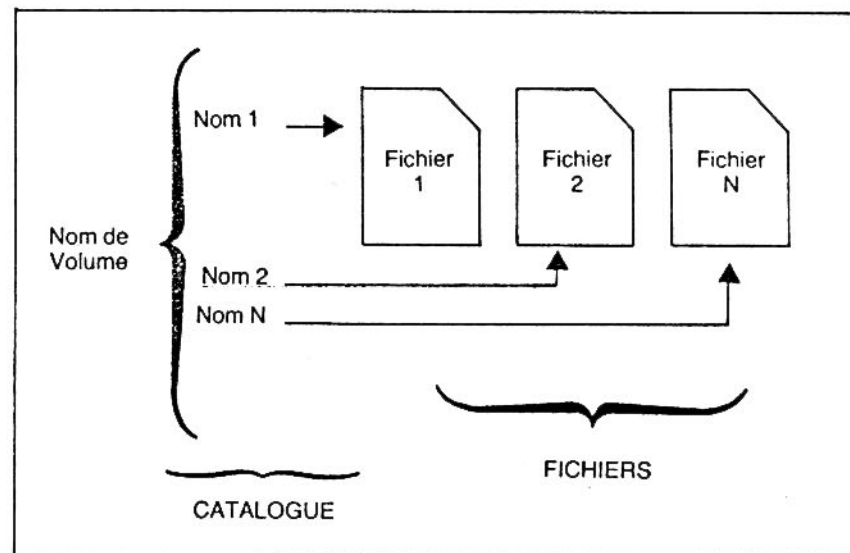
C'est le T.DOS qui fait la liaison entre cette disquette logique et la disquette physique.

1) - La première chose qui vient à l'esprit, est qu'il serait pratique de différencier les disquettes. Le plus simple étant de leur donner un nom, T.DOS permet cela. C'est la notion de **nom de volume**. Cela consiste en une chaîne de caractères (de huit caractères au maximum) nom de volume ou nom de disquette.

2) - Vous allez implanter des fichiers sur votre ou vos disquettes. Il faut également pouvoir les distinguer entre eux. C'est la notion de **noms de fichiers**. T.DOS vous permet de donner un nom à vos fichiers.

De plus, une sous-différenciation est possible grâce aux types (BIN, BAS, TXT, DAT, CMD). Deux noms identiques sont tolérés s'ils n'ont pas le même type.

3) - C'est excellent d'avoir des noms, mais encore faut-il pouvoir en obtenir une liste. C'est la notion de **catalogue**. T.DOS gère un catalogue de tous les fichiers et vous le "liste" si vous en faites la demande.



DISQUETTE LOGIQUE

4) - Une dernière notion : si vous êtes l'heureux possesseur de plusieurs **unités** de lecteurs de disquettes. (Un module JASMIN peut en contenir 2) L'électronique du JASMIN permet de gérer jusqu'à 4 unités.

Il faut connaître et reconnaître chaque unité séparément. C'est le rôle du **numéro d'unité disque** (de 1 à 4).

5) Numéro logique NLU de fichier :

Si nous regroupons les notions 1, 2 et 4 précédentes vous vous apercevrez tout de suite que pour accéder à un fichier, il vous faut connaître : le nom du volume, le nom du fichier, son type et accessoirement son numéro d'unité. C'est beaucoup, mais nécessaire. Par contre ce qui n'est pas nécessaire c'est de répéter tout cela à chaque fois que l'on veut accéder à ce fichier.

T.DOS permet de n'effectuer cette association qu'une seule fois par fichier.

Le principe est d'affecter un numéro à l'ensemble : .

-Numéro d'unité - nom de fichier - TYP

C'est ce qu'on appelle le numéro logique du fichier. C'est pratique et puis de toute façon obligatoire....

CHAPITRE II - UTILISATION DU T.DOS

GENERALITES SUR LES DISQUETTES

D'entrée nous allons vous donner les quelques règles à respecter pour la syntaxe ainsi que les conventions prises pour décrire chaque commande.

- 1) Chaque commande T.DOS commence impérativement par le caractère «!» .
- 2) Les noms de fichiers commencent obligatoirement par un caractère alphabétique «A à Z» et ne doivent pas comporter d'espace en leur milieu.
! SAVE "1.TURLUTU" ! SAVE "1.4TUR LU"
 oui incorrect
- 3) De même les commandes syntaxiquement groupées ne doivent pas comporter d'espace.
! SAVE "1.TITI.BAS" ! SAVE "1.TITI. BAS"
 oui incorrect

Une commande syntaxiquement groupée est une commande dont les paramètres sont séparées par des «.» (points).

Règle de description syntaxique

- 1) Les parties situées entre parenthèses sont optionnelles et peuvent prendre une valeur par défaut qui est toujours clairement indiquée.
- 2) Les parties soulignées par un trait continu sont obligatoires et ne peuvent être omises.
- 3) Les parties soulignées par un trait discontinu sont obligatoires de façon conditionnelles. La condition de présence obligatoire est toujours indiquée clairement dans le texte décrivant le paramètre.
- 4) Les parties où il n'y a rien d'écrit ne signifie rien (ne cherchez pas à lire entre les lignes !).

Règle sur les noms de fichiers :

T.DOS accepte des fichiers ayant le même nom à condition qu'ils n'aient pas le même type ou qu'ils ne résident pas sur la même unité disque.

1.PAPA.BIN et PAPA.DAT sont deux fichiers différents. Il en est de même pour 1.PAPA.BIN et 2.PAPA.BIN

A - OPERATIONS SUR LES DISQUETTES :

A - 1 système mono disquette et multidisquette

Sous vos yeux émerveillés, un lecteur JASMIN attend que le gentil Maître que vous êtes, se décide à lui donner sa chance de vous montrer de quoi il est capable. Si vous faites n'importe quoi il en fera de même. Le seul problème, c'est que le lecteur, lui s'en moque, mais vous, peut être pas. Alors, suivez nos conseils, vous improviserez plus tard, après avoir pris les précautions d'usage.

Montez maintenant la disquette T.DOS dans le lecteur, face A vers le haut et appuyer sur le bouton «BOOT».

Le système T.DOS est chargé. La disquette que vous avez dans le lecteur est la disquette MAITRE (MASTER). Elle a pour nom de volume T.DOS et quelques utilitaires (pour vérifier, frapper la commande !CAT""

A partir de là, vous pouvez commencer à travailler si vous le désirez (c'est à dire créer des fichiers et les utiliser).

Vous pouvez d'abord opérer comme suit : **Formatez, Initialisez, et donnez un nom à une autre disquette.**

Voici les commandes générales, dans le cas d'un monolecteur, le n° DISQ est bien évidemment égal à 1.

A - 2 Formattage : commande !FORMAT"

* Syntaxe : !"FORMAT"

C'est un utilitaire Basic. C'est un programme conversationnel, suivez ce qu'il vous demande, mais ne pas oublier de retirer la disquette.

* Action : Formattage de l'unité de Disque, et ensuite le programme INIT si vous le désirez

Remarque : lorsque vous tapez !"FORMAT", la disquette Maitre doit être dans le lecteur.

A - 3 Initialisation : commande !INIT

- * Syntaxe : !INIT "(N° DISQ.) NOMVOLUME"
- * Paramètres : N° DISQ = Numéro de l'unité disque (de 1 à 4)
NOMVOLUME : Nom de volume (de la disquette) que vous voulez initialiser 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué à droite de 8 caractères.

* Action : Cette commande initialise le catalogue.
Attention : Dans le cas d'une disquette contenant des fichiers il y a **perte de ces fichiers.**

Voilà, vous avez effectué ces opérations sur toutes les disquettes neuves en votre possession. C'est alors que, illumination de votre part, vous vous dites il serait bien que ces disquettes puissent être «Bootstrappées et contenir le T.DOS. et bien, puisque vous y tenez tant, faites le.»

Voici la commande, mais attention, cette commande comporte une restriction....

Lisez la.

A - 4 Copie du T.DOS : Commande !MASTER

- * Syntaxe : !MASTER "(N°DISQ.) NOMTDOS"
- * Paramètres : N° DISQ = Numéro de l'unité disque (1 à 4)
NOMTDOS : Nom que l'on donne au système d'exploitation disque 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué de 8 caractères.
- * Action : Cette commande permet de rendre une disquette MAITRESSE (un disque MAITRE), en recopiant dessus le système T.DOS. Cette disquette devient alors "Bootstrappable".

A - 5 Nomination : commande ! DNAME

- * Syntaxe : ! DNAME "(N° DISQ.) NOMVOLUME"
- * Action : changer le nom d'une disquette sans effacer les fichiers existants.
- * Paramètres : N° DISQ = Numéro de l'unité disque (1 à 4) Valeur par défaut 1.

NOMVOLUME : Nom que vous voulez donner à votre disquette. 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué à droite de 8 caractères. Le premier caractère doit être alphabétique (de A à Z).

- * Action : Cette commande permet de donner un nom à une disquette ou de renommer une disquette (changement de nom).

Exemple : ! DNAME "ESCLAVE"

A - 6 Liste du catalogue sur écran ! CAT

- * Syntaxe : !CAT "(NODISQ.)"
!CAT "" pour le disque 1
!CAT "2." pour le disque 2
- * Fonction : liste du catalogue complet des fichiers contenus sur la disquette.

Vous verrez par exemple :
VOLUME : NOMVOLUME

```
L TDOS 1-2 .SYS S 62 SECTORS
U FORMAT .BAS S 13 SECTORS
U FICHES .DAT D 22 SECTORS
```

355 SECTORS FREE

La première lettre L ou U indique si le fichier est respectivement verrouillé ou non par la fonction !LOCK. Ensuite suivent le nom du fichier et son type ou extension.

Après l'extension, une lettre isolée S ou D indique si le fichier est séquentiel ou à accès direct.

La taille du fichier est exprimée en nombre de secteurs. Chaque secteur contient 256 octets.

A la fin vous avez l'espace disponible du disque.

Vous pouvez interrompre momentanément le listage en appuyant sur la barre "espace". Ensuite à chaque pression, vous imprimez une ligne.

Pour arrêter définitivement cette instruction, tapez "Return".

A - 7 Liste du catalogue sur imprimante !LCAT

Fonctionne comme !CAT mais liste sur imprimante.

A - 8 Copie d'une disquette entière !"BKP"

* Syntaxe : ! "BKP"

* Action : Copie intégrale d'une disquette sur une autre disquette. (C'est un utilitaire conversationnel qui se trouve sur la disquette "Maître" et qui se charge en mémoire à chaque appel). Répondez aux questions que vous pose le programme.

La disquette "Émettrice" est celle que l'on peut dupliquer. La disquette "Réceptrice" est celle qui va devenir le duplicata.

Lorsque vous avez deux lecteurs, le transfert se fait sans manipulation. Si vous n'avez qu'un seul lecteur, vous pouvez suivre scrupuleusement les indications du programme. La taille mémoire de l'ordinateur étant plus petite que celle d'une disquette, l'opération se fera en plusieurs fois.

Vous aurez à permuter plusieurs fois les disquettes.

Veillez à ne pas vous tromper entre la disquette "émettrice" et "réceptrice".

A - 9 Montage d'un volume ! MOUNT

Vous êtes en possession de plusieurs unités de lecteurs (maximum 4), d'une disquette supportant le système TDOS et d'autres disquettes neuves. Vous effectuez tous vos branchements, la disquette T.DOS dans le lecteur n° 1, les disquettes formatées et initialisées dans les autres unités.

"Bootstrapez" la disquette T.DOS pour charger le système en mémoire.

Si votre système n'a pas été informé pour être utilisé en "MULTI", il faut le prévenir par l'instruction ! MOUNT.

Comme pour le système mono-lecteur, vous pouvez recopier le T.DOS sur d'autres disquettes. La procédure est identique sauf que vous n'avez pas à effectuer de changement de disquette (attention au paramètre N° DISQ.).

Lorsque vous avez configuré votre système, c'est à dire "Monté" tous les lecteurs, refaites un !MASTER, ainsi la configuration définie reste valable chaque fois que vous refaites un "BOOT".

* Syntaxe : !MOUNT "N° DISQ."

* Paramètre : N° DISQ. = Numéro de l'unité disque (1 à 4)

* Action : Cette commande a pour but de faire connaître au T.DOS en mémoire, qu'il existe d'autres unités de lecteur que le n° 1, connectées ou "Montées". Cette fonction vous permet de configurer votre système.

A - 10 Démontage d'un volume !DEMOUNT

Cette commande indique au T.DOS en mémoire que la configuration est modifiée, et qu'il y a démontage d'une unité de lecteur.

A - 11 Recopie de fichiers - commande !COPY

Un système T.DOS serait incomplet s'il ne permettait pas la duplication de fichier. Ainsi T.DOS a une commande pour cet usage.

* Syntaxe : !COPY "(N° DISQ1.) NOMA (.TYP) =(N°DISQ2.) NOMB (.TYP2)"

* Paramètres : N°DISQ1 : Numéro de l'unité disque émetteur
Valeur par défaut 1

NOMA : Nom de fichier émetteur ou à recopier. 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué à droite à 8 caractères. Si le nom existe, le fichier est écrasé, sinon il est créé.

TYP1 : BIN
BAS
TXT
ARY
DAT
CMD

Valeur par défaut : BAS

N°DISQ2 : Numéro de l'unité réceptrice (de 1 à 4)
Valeur par défaut : 1.

NOMB : Nom du fichier récepteur, 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué à droite de 8 caractères.

TYP2 : BIN
BAS
TXT
ARY
DAT
CMD

Valeur par défaut : type de l'émetteur.

*** Remarque :**

Lors de la recherche du fichier à recopier, si le paramètre TYP1 est présent, les fichiers ayant pour nom NOMA mais dont le type ne correspond pas à TYP1, ne seront pas considérés comme le fichier à recopier.

Exemple :

!COPY "TOTO.BIN = 2.TATA" copie de l'unité 1 à 2

!COPY "TOTO.BIN = TITI.CMD" copie de la même unité.

A - 12 Copie de fichier binaire : commande !"COPY1"

C'est un utilitaire conversationnel qui vous permet de recopier un fichier binaire sur une autre disquette, au cas où vous n'avez qu'un seul lecteur.

B - TRAITEMENT SUR TOUT FICHER

B - 1 Sauvegarde !SAVE

De la même manière que vous pouvez effectuer des sauvegardes sur cassettes, T.DOS donne cette possibilité. La fonctionnalité est identique.

- * Syntaxe : !SAVE“(N°DISQ.)NOMFICH.(.TYP)(,AD,AF)”
- * Paramètres : N° DISQ : Numéro de l'unité disque de 1 à 4
Valeur par défaut : 1

NOMFICH : Nom du fichier sur 8 caractères alphanumériques obtenus par troncature à droite. Le premier caractère doit être obligatoirement alphabétique (A à Z).

TYP : BAS : indique la sauvegarde d'un programme écrit en Basic.

BIN : indique la sauvegarde en binaire d'une zone de mémoire délimitée par les adresses AD et AF.

Valeur par défaut : BAS.

Si TYP : BIN
TXT
DAT
CMD
SCR

} : AD,AF : Adresse de début, adresse Fin.

Les valeurs peuvent être en décimal ou en hexadécimal précédées du caractère " # ", AD et AF peuvent être des variables BASIC.

Exemple : 10 AD = # 6000

20 AF = # 8000

30 A\$ = "TOT.BIN,AD,AF"

40 !SAVE A\$

ou !SAVE"TOT.BIN, # 6000, # 8000

Action : Sauvegarde par la disquette portée par l'unité n° DISQ du programme BASIC en mémoire si TYP = BAS ou de la zone nécessaire définie par AD et AF incluses si TYP = BIN, TXT, DAT, SCR, CMD. Le nom du fichier ainsi créé est NOMFICH.

Si un fichier de même nom et de même type existe déjà dans le catalogue, il y a écrasement du précédent fichier par le nouveau, sauf si le fichier est protégé par LOCK.

B - 2 Chargement !LOAD

- * Syntaxe : !LOAD"(N° DISQ.) NOMFICH (.TYP) (,AD)"
- * Paramètres : N°DISQ : Numéro de l'unité disque de 1 à 4
Valeur par défaut : 1

NOMFICH : Nom du fichier à charger sur 8 caractères obtenus par troncature à droite. Le premier caractère doit être obligatoirement alphabétique (de A à Z).

TYP : .BIN : Indique le chargement d'une zone mémoire. Si le paramètre AD est présent, le chargement s'effectue à partir de cette adresse, dans le cas contraire l'adresse du début est celle donnée lors de la commande !SAVE.

SCR
TXT . }
DAT : } Comme BIN
CMD : }

Valeur par défaut : BAS

Si TYP = BIN AD : Adresse à partir de laquelle doit être chargée la zone mémoire contenue dans le fichier.

La Valeur décimale ou hexadécimale AD peut être une variable BASIC.

- * Action : Chargement à partir de la disquette portée par l'unité N°DISQ. du programme basic si TYP= BAS, ou de la zone mémoire si TYP = BIN, TXT, DAT ou CMD.

Dans ces derniers cas, si le paramètre AD est présent le chargement s'effectue à partir de cette adresse. (possibilité de décalage de zones).

Après exécution de cette instruction, l'adresse de début se trouve en #C062 et #C063 en RAM parallèle et l'adresse de fin en #48F et #490.

B - 3 Chargement et exécution : !

- * Syntaxe : !"(N°DISQ.)NOMFICH (.TYP)".
- * Paramètres : N° DISQ. = Numéro de l'unité disque de 1 à 4
Valeur par défaut : 1

NOMFICH : Nom de fichier à charger sur 8 caractères obtenus par troncature à droite. Le premier caractère doit être obligatoirement alphabétique (de A à Z).

TYP : BAS : Indique le chargement d'un programme écrit en basic.

BIN ou CMD : Indique le chargement d'un programme écrit en langage machine, le chargement s'effectue à partir de l'adresse du début donnée lors de la commande !SAVE.
Valeur par défaut : BAS.

- * Action : Chargement à partir de la disquette portée par l'unité N°DISQ du programme Basic, si TYP=BAS ou du programme en code machine si TYP=BIN ou CMD et l'exécute.
Cette instruction permet aussi des lancements automatiques d'un programme à partir d'un autre programme.

Exemple :

!"FORMAT" ou !"TKD" ou !"TOT.CMD"

B - 4 Transfert d'un fichier de cassette à disquette sans déprotéger : !"TKD"

- * Syntaxe : !"TKD"
- * Action : Transférer un fichier d'une cassette sur la disquette. C'est un utilitaire conversationnel qui se charge en mémoire à chaque appel. Suivez les indications du programme. Si vous ne connaissez pas le nom du programme sur cassette, mettez "".

Dans ce cas, le premier fichier rencontré est transféré. Lorsque le programme vous demande le nom de fichier sur disquette, vous mettez le nom que vous voulez sur 8 caractères maximum, commençant par un caractère alphabétique.

Si le nom sur cassette commence par un caractère autre qu'alphabétique, ou avec un "blanc" de séparation, vous devez changer le nom sur disquette, sinon vous ne pourrez plus y accéder.

Cet utilitaire est créé dans le but de vous faciliter la sauvegarde de vos programmes ou fichiers de votre cassette sur votre disquette et non pirater les logiciels protégés. Il lit la cassette selon le codage standard d'ORIC. Si votre fichier a été codé d'une autre manière, cet utilitaire ne pourra le lire. Si votre programme a été enregistré en plusieurs tronçons, le transfert se fera tronçon par tronçon. C'est à vous d'assurer ensuite le lien entre ces tronçons. Pour arrêter l'opération en cours, faites un RESET à chaud par le bouton en dessous de l'ORIC ou faites un RESET général.

B - 5 Recherche de fichier !SEARCH

- * Syntaxe : !SEARCH"(N°DISQ.) NOMFICH.TYP"
- * Action : Rechercher dans le catalogue le fichier de nom NOMFICH.TYP. Le type est obligatoire. En mode "commande", si le fichier existe, le message "EXISTING FILE!" est imprimé, sinon "FILE NOT FOUND". En mode programmé, l'impression des messages interrompt le programme comme si c'était une erreur. Cette instruction peut être utilisée en combinaison avec IERSET. Dans ce cas, se trouve en # 489, 1 si le fichier existe sinon 7.

Exemple : 5 REM RECHERCHE DE FICHER

```
10 !ERSET
20 INPUT "NOM DE FICHER"; A$
30 B$ = A$ + ".BAS"
40 !SEARCH B$
50 !IF PEEK ( # 489) = 1 THEN PRINT
  B$, "EXISTE"
60 B$ = A$ + ".BIN"
70 !SEARCH B$
80 !IF PEEK ( # 489) = 1 THEN
  PRINT B$, "EXISTE"
```

etc...

B- 6 Changement de nom !RENAME

Changement de nom d'un fichier - commande ! RENAME

- * Syntaxe : !RENAME "(N° DISQ.) NOMFICA (.TYP) =NOMFICB (.TYPB)"
- * Paramètre : N° DISQ. = Numéro de l'unité qui supporte le fichier à renommer.
Valeur de 1 à 4
Valeur par défaut : 1

NOMFICA : Nom du fichier à renommer de 1 à 8 caractères (avec troncature à droite pour les tailles supérieures). Le premier caractère obligatoirement alphabétique.

TYPE : ARY
BAS
BIN
TXT
DAT
CMD
Valeur par défaut : BAS

NOMFICB : Nouveau nom du fichier. De 1 à 8 caractères alphanumériques troncature à droite pour plus de 8 caractères. Le premier caractère est obligatoirement alphabétique.

- * Action : Cette commande permet de changer le nom (pas le type) d'un fichier et ce pour tous les fichiers sauf les fichiers systèmes.

B - 7 Protection d'écriture !LOCK

- * Syntaxe : !LOCK"(N° DISQ.) NOMFICH.TYP"
- * Paramètres : habituels sans plus.
- * Action : Interdire l'écriture dans un fichier désigné, de même que son effacement (!DEL) ou son écrasement (!SAVE)
Par sécurité le type ne peut être omis.

B - 8 Déprotection d'écriture !UNLOCK

- * Syntaxe : !UNLOCK"(N°disq.) NOMFICH.TYP"
- * Paramètres : habituels sans plus.

- * Action : Enlever l'interdiction de l'écriture dans le fichier désigné ou de son effacement.
Fonction inverse de !LOCK
Par sécurité le type ne peut être omis.

B - 9 Concaténation d'un fichier Basic à la suite d'un autre fichier Basic !MERGE

- * Syntaxe : !MERGE "(NODISQ.) NOMFICH"
- * Action : Adjonction du fichier NOMFICH.BAS à la suite du fichier BASIC existant en mémoire, tout en gardant les variables existant en mémoire. Ce qui permet d'effectuer des chaînages de programmes en continuant leur exécution. Le programme appelé doit être en BASIC. Vous devez veiller à ce que tous ses numéros de lignes soient supérieurs à ceux existant en mémoire centrale. Nous ne garantissons pas du résultat autrement.

Argument : Le type du fichier peut être omis, car il est d'office . BAS.

B - 10 Destruction de fichier !DEL

- * Syntaxe : !DEL "(NODISQ.) NOMFICH.TYP"
- * Action : Effacer le fichier NOMFICH.TYP, l'enlever du catalogue et libérer la place qu'il occupait sur la disquette.

Attention : pour éviter l'effacement par erreur, le type du fichier est obligatoire.

Exemple : !DEL "TOTO.BAS"

B - 11 Effacement à partir de la ligne numéro N !CUT N

- * Syntaxe : !CUT N
- * Paramètre : N : Numéro de ligne
- * Action : Efface les lignes de programme de numéro N à la fin.

Cette instruction, utilisée avec !MERGE, vous permet de faire tourner des programmes qui ne tiendraient pas dans les 42 K octets utilisateurs. Vous pouvez garder un morceau de programme fixe en

mémoire centrale, appelé "Noyau", les autres morceaux seront appelés à tour de rôle par !MERGE et effacés pour libérer la place par !CUT. Cette instruction conserve la valeur des variables en cours.

C - TRAITEMENT D'ERREUR

C-1 Empêche l'interruption en cas d'erreur du DOS !ERSET

- * L'exécution du programme en cours ne s'interrompt pas à la ren contre d'une erreur d'utilisation du T.DOS, et continue sur la pro- chaine instruction. A la place du message d'erreur, le code d'erreur est mis en # 489, que vous pouvez consulter par un !PEEK (# 489) par exemple. Cette instruction est très intéres- sante lorsqu'elle est utilisée avec !ERR GOTO, voir un exemple au chapitre correspondant.

C-2 Autorise l'interruption normale en cas d'erreur d'utilisa- tion du DOS !EROFF

- * Syntaxe : !EROFF.
- * Action : Autorise l'interruption d'un programme et l'impression du message d'erreur en cas d'erreur d'utilisation du T.DOS. C'est la fonction inverse de !ERSET.

C - 3 En cas d'erreur d'utilisation du DOS, continue à la ligne N° !ERR GOTO NN

- * Syntaxe : !ERR GOTO NN
- * Action : C'est l'équivalent de "ON ERROR GOTO" de certain BASIC, à utiliser avec !ERSET. Quand le programme arrive à cette instruction, il regarde s'il n'y avait pas une erreur d'utilisation du T.DOS, si c'est le cas, il va à la ligne NN, sinon il continue à l'instruction suivante.

Exemple :

```
10 REM chargement d'un FICHIER
20 !ERSET
30 INPUT "NOM DE FICHIER" ; A$
40 !LOAD A$
50 !ERR GOTO 100
60 PRINT "FICHIER";A$; "CHARGE"
70 INPUT "POUR CONTINUER TAPEZ "O : " ; O$
80 IF O$ = "O" THEN 30
90 END
100 ER = PEEK ( # 489)
110 IF ER = 7 THEN 200
120 PRINT "RECOMMENCER"
130 GOTO 30
200 PRINT "CE FICHIER N'EXISTE PAS"
210 GOTO 120
```

Remarque :

Le code d'erreur 7 correspond à "FILE NOT FOUND". Ce traitement d'erreur ne s'occupe pas des erreurs du BASIC. La table des codes d'erreur est donnée en Annexe A4.

D - FICHIERS DE DONNEES DE TYPE. DAT

Généralités.

Maintenant que vous savez préparer vos disquettes, sauvegarder et charger vos programmes, il est temps de passer à un stade supérieur : l'utilisation des fichiers autres que les fichiers binaires, basics ou commandes.

Ce sont des fichiers de données, caractérisés par le type .DAT (pour DATA). Ils peuvent être à accès séquentiel ou direct.

L'utilisation de tels fichiers suit toujours le même protocole suivant.

1) - **Création**: si le fichier n'existait pas, il faut le créer, et l'ouvrir par l'instruction !CREATE.

2) - **Ouverture** : si le fichier existait déjà sur disquette, il faut l'ouvrir. C'est comme si l'on ouvrait un compte, un dossier, un livre, etc...

Chaque fois que vous créez et ouvrez ou ouvrez simplement un fichier, vous devez lui donner en même temps un numéro, appelé numéro logique du fichier.(NLU)

C'est comme si on donnait un numéro de compte ou de dossier. Après, jusqu'à la fermeture du fichier, il n'est connu que sous son numéro logique.

Vous créez par exemple un fichier en lui donnant le nom "ESSAI". Il sera sauvegardé sur disque sous le nom "ESSAI".DAT, mais dans l'ordinateur, il sera ouvert et connu par son numéro logique que vous lui avez donné.

Exemple : !CREATE "ESSAI,S =5"

Vous avez créé un fichier ESSAI.

Il est séquentiel, vous l'avez indiqué par S. Et il a le numéro logique 5.

Vous pouvez avoir jusqu'à 16 fichiers ouverts en même temps.

Le numéro logique est un entier de 1 à 125.

Le numéro zéro est interdit. Il a une signification bien particulière pour les instructions !CLOSE, !REWIND, !APND.

3) - Lecture et écriture : Une fois qu'un fichier est ouvert, vous pouvez écrire dessus et relire ce qui a été écrit.

C'est là que se distinguent deux types de fichiers :

le fichier à accès séquentiel

et le fichier à accès direct.

Comme leur nom l'indique, ils sont différents par leur façon d'accéder aux informations sauvegardées sur disquette.

Ces deux types de fichiers mémorisent des chaînes de caractères représentés sous leur forme de valeur ASCII.

Dans un fichier à accès séquentiel, les informations que vous écrivez dedans sont mises l'une juste derrière l'autre.

Vous relisez, l'une derrière l'autre, les informations dans l'ordre où vous les avez écrites.

Une bande magnétique est un très bon exemple de fichier séquentiel. Si vous n'avez pas repéré à l'avance où se trouve enregistrée votre information sur la bande, vous êtes obligé de lire la bande depuis le début, jusqu'au moment où vous trouvez votre information.

Un tel fichier est simple mais devient très vite peu pratique lorsque vous avez des informations que vous voulez changer souvent, surtout quand elles ne sont pas situées l'une derrière l'autre.

Le fichier à accès direct aléatoire est plus structuré et va donc vous permettre d'accéder directement à une zone d'informations voulue, sans avoir à lire tout ce qui se trouvait avant. Chaque zone d'informations est appelée "enregistrement".

Chaque enregistrement peut être composé de plusieurs rubriques.

Ce fichier est, en fait, une suite d'information (en l'occurrence des mots). Chaque élément d'information est dénommé "Rubrique".

Chaque rubrique commence par un octet qui contient sa longueur en nombre d'octets, suivi par le contenu et fini par un octet " # FF".

La longueur maximale d'une rubrique est de 256 octets. La longueur maximale réelle de l'information est alors de 254 octets.

Un fichier séquentiel est donc formé d'une suite de rubriques. Ces rubriques peuvent être de longueurs variables.

Un fichier à accès direct aléatoire est composé d'enregistrement de longueur fixe. Chaque enregistrement est en fait un petit fichier séquentiel qui comporte des rubriques de longueurs variables. La longueur maximale d'un enregistrement est 256 octets. Il va de soi que la somme des longueurs des rubriques d'un enregistrement ne doit pas dépasser la longueur de ce dernier.

Pour une variable réelle, le système réserve automatiquement 17 octets et pour une variable entière (%), 8 octets.

Pour lire ou écrire dans un fichier à accès direct, il vous suffit de préciser le numéro logique du fichier concerné, le numéro d'enregistrement dans lequel vous voulez travailler.

Ce type de fichier pratique d'utilisation, est indispensable pour des applications où l'on a besoin d'accéder, dans un ordre quelconque, rapidement aux informations.

4)- Fermeture :

Quand vous classez une affaire, vous fermez le dossier. Quand vous avez fini de lire un roman, vous fermez le livre.

Pour un fichier, c'est pareil, quand vous n'avez plus besoin d'un fichier, vous devez le fermer. Ce qui a pour effet de sauver certains paramètres du fichier et libérer le numéro logique correspondant.

Attention :

Les fichiers à accès séquentiel ou à accès direct ont des paramètres qui sont bien particuliers à la configuration du moment de la disquette, ils ne peuvent donc pas être copiés par fonction !COPY. Vous devez soit utiliser la fonction !BKP pour recopier tout le disque dans la même configuration, soit recopier par un programme, enregistrement par enregistrement ou rubrique par rubrique.

D - 2 Protection d'écriture !WL

* Syntaxe : !WL NLU

* Paramètre : NLU = Numéro logique d'unité du fichier déjà ouvert.
 $1 \leq NLU \leq 255$
NLU = 0 est interdit.

* Action : Protège contre toute tentative d'écriture sur le fichier. Cette protection ne s'étend pas jusqu'à l'effacement ou l'écrasement du fichier par des instructions !DEL, !SAVE etc...
Pour protéger contre ces dernières, il faut utiliser !LOCK.

D - 3 Déprotection d'écriture !WUL

* Syntaxe !WUL NLU

* Paramètre : NLU = Numéro logique du fichier déjà ouvert.
 $1 \leq NLU \leq 255$
NLU : 0 est interdit.

- * Action : fonction inverse de !WL, permet l'écriture sur le fichier NLU

D - 4 Ouverture de fichier !OPEN

- * Syntaxe : !OPEN "(N°DISQ.) NOMFICH = NLU"

Paramètre : NLU : Numéro logique qui sera affecté au fichier de nom NOMFICH.DAT. c'est une valeur décimale, hexadécimale ou une variable.
1 = < NLU = < 255

TYPE : Le type du fichier sera mis d'office à .DAT, quelque-soit le type que vous avez indiqué.

- * Action : Cette commande ouvre le fichier existant NOMFICH.DAT et lui associe le numéro logique NLU.

D - 5 Fermeture du fichier !CLOSE

- * Syntaxe : !CLOSE NLU

- * Paramètre : NLU : Numéro logique du fichier à fermer, Valeur décimale, hexadécimale ou variable.
1 = < NLU = < 255
Si NLU = 0, tous les fichiers ouverts sont concernés.

- * Action : Fermeture du fichier de numéro logique NLU ou fermeture de tout fichier ouvert si NLU = 0. A partir de l'exécution de cette commande, l'utilisateur n'a plus accès aux informations contenues dans le fichier, que par le numéro logique NLU.

E - FICHIERS A ACCES SEQUENTIEL

E - 1 Création d'un fichier !CREATE

- * Syntaxe : !CREATE "(N°DISQ.) NOMFICH, FTYPE = NLU"
- * Paramètres : NLU = Numéro logique qui sera affecté au fichier. C'est une valeur décimale, hexadécimale ou une variable.
1 = < NLU = < 255

- * TYPE : Le type du fichier sera mis d'office à .DAT, quelque-soit le type que vous avez indiqué.

- * FTYPE = S pour fichier à accès séquentiel.

- * Action : Cette commande crée le fichier NOMFICH en accès séquentiel sur disque et l'ouvre en mémoire centrale en lui associant le numéro logique NU.

E - 2 Ouverture de fichier !OPEN

- * Syntaxe : !OPEN "(N°DISQ.) NOMFICH = NLU"

- * Paramètre : NLU = Numéro logique qui sera affecté au fichier de nom NOMFICH.DAT
C'est une valeur décimale, hexadécimale ou une variable.
1 = < NLU = < 255

- * Type : Le type du fichier sera mis d'office à DAT, quelque soit le type que vous avez indiqué.

- * Action : Cette commande ouvre le fichier existant NOMFICH.DAT et lui associe le numéro logique NLU.

E - 3 Fermeture du fichier !CLOSE

- * Syntaxe : !CLOSE NLU

- * Paramètre : NLU = Numéro logique du fichier à fermer
Valeur décimale ou hexadécimale ou variable.
1 = < NLU = < 255
Si NLU = 0, tous les fichiers ouverts sont concernés.

- * Action : Fermeture du fichier de numéro logique NLU ou fermeture de tout fichier ouvert si NLU = 0. A partir de l'exécution de cette commande, l'utilisateur n'a plus accès aux informations contenues dans le fichier, par le numéro logique NLU.

E - 4 Ecriture !WRITE

- * Syntaxe : !WRITE NLU <liste des variables>
ou !WRITE NLU : REM <liste des variables>
pour version ORIC 1

- * Paramètre : NLU = numéro logique de fichier
1 = < NLU = < 255

Liste des variables : Suite de variables séparés par des “,”. Tous les types de variables sont acceptés sauf la variable chaîne en tableau comme A\$ (5,4).

Attention : Lorsqu'une variable chaîne est utilisée dans la liste des variables, elle doit être utilisée ou exister déjà avant. Sinon il faut la définir par exemple par A\$ = “”, avant d'utiliser cette instruction.

- * Action : Cette commande permet l'écriture dans un fichier si ce dernier n'est pas verrouillé par !WL.
Une variable flottante occupe 17 octets.
Une variable entière occupe 8 octets.
Une variable chaîne a une longueur variable.

Exemple : A\$ = “” : I = 5 : NLU = 120 : B% = 400 : C\$ = “TOTO”
!WRITE NLU I,A,B%,C\$.
Chaque variable occupe une rubrique.

Vous avez la possibilité de réécrire sur une rubrique existante, donc la modifier. Mais prenez garde, une fois qu'une rubrique d'un fichier séquentiel a été créée par une première écriture, sa longueur est fixée. Tout ce que vous allez écrire dessus va garder automatiquement la même longueur. C'est le T.DOS qui s'occupe de cette opération et vous évite d'écraser par mégarde la rubrique suivante.

Résultat : Lorsque votre nouvelle rubrique est trop longue, elle sera tronquée par la droite. Si elle est plus courte, la suite est remplie par des “espaces”. Chaque fois que vous avez écrit une rubrique, vous êtes positionné pour accéder à la rubrique suivante.

E - 5 Lecture !TAKE

- * Syntaxe : !TAKE NLU <Liste des variables>
ou !TAKE NLU : REM <Liste des variables>
pour Oric 1
- * Paramètre : NLU = Numéro logique du fichier que l'on veut lire.
1 = < NLU ≤ 255

<Liste des variables> Suite des variables BASIC que l'on veut lire, séparées par des “,”. Tous types de variables sont acceptés sauf la variable chaîne en tableau comme A\$ (5,4).

Attention : Lorsqu'une chaîne est utilisée dans la liste des variables, elle doit être utilisée ou exister déjà avant. Sinon il faut la définir par A\$ = “”, avant d'utiliser cette instruction.

- * Action : La commande !TAKE effectue la lecture d'une suite de rubriques suivant la liste de variables. Vous ne pouvez pas lire plus loin que ce que vous avez écrit. Chaque fois que vous avez lu une rubrique, vous êtes positionné pour accéder à la rubrique suivante.

E - 6 Positionnement en début de fichier !REWIND

- * Syntaxe : !REWIND NLU
- * Paramètre : NLU = Numéro logique du fichier à accès séquentiel déjà ouvert.
1 = < NLU = < 255
NLU peut être une variable.
Lorsque NLU = 0, l'opération s'effectue sur tous les fichiers séquentiels ouverts.
- * Action : Positionnement en début du fichier séquentiel NLU. Lorsqu'un fichier vient d'être ouvert, il est positionné sur la rubrique où il était la dernière fois, avant d'être fermé. C'est une des particularités intéressantes du T.DOS. Ce qui vous permet d'interrompre une cession, et de la reprendre au même point où vous l'avez laissée. Si vous voulez tout reprendre depuis le début, n'oubliez pas de faire !REWIND 0 après avoir ouvert les fichiers. Le terme REWIND qui veut dire rebobinage, vient du fait qu'avant l'arrivée des disques, les fichiers étaient sur bandes.

E - 7 Positionnement en fin de fichier !APND

- * Syntaxe : !APND NLU
- * Paramètre : NLU : Numéro logique du fichier à accès séquentiel déjà ouvert.
1 = < NLU = < 255
NLU peut être variable.
Lorsque NLU = 0 l'opération. s'effectue sur tous les fichiers séquentiels ouverts.
- * Action : Positionnement en fin du fichier séquentiel NLU. Ce qui va vous permettre de rajouter d'autres rubriques à la suite.

Exemple : 10 ! OPEN "ESSAI = 5"
 20 ! APND 5
 30 A = 55 : C\$ = "BONJOUR"
 40 !WRITE 5' A,C\$

Lorsque vous rajoutez les rubriques à la fin d'un fichier séquentiel, ces rubriques seront écrites pour la première fois, leur longueur est déterminée comme suit :

Une variable flottante occupe 17 octets
 Une variable entière occupe 8 octets
 Une variable chaîne a une longueur variable.

E-8 Indication du N° de rubrique courante du fichier et du nombre total de rubriques !WHERE

- * Syntaxe : ! WHERE NLU
- * Paramètre : NLU = Numéro logique du fichier ouvert
 $1 \leq NLU \leq 255$
 NLU = 0 est interdit.
- * Action : Mettre à l'adresse # 48 D, le numéro de rubrique en cours du fichier NLU et en. # 48 F le nombre total de rubriques du fichier. Ex. : pour obtenir le nombre total de rubriques faites A = DEEK (# 48 F)

E-9 Incrémentation du pointeur de rubrique !JUMP

- * Syntaxe : ! JUMP NLU, NI
- * Paramètre : NLU = Numéro logique du fichier déjà ouvert.
 $1 \leq NLU \leq 255$
 NLU = 0 est interdit.
 NI = Nombre d'incrémentations du pointeur
 $1 \leq NI \leq \# \text{FFFF}$
- * Action : Sauter NI rubriques et se positionner NI rubriques plus loin. Si vous êtes à la N0^{ième} rubrique, !JUMP NLU, NI vous positionne à la N0 + NI^{ème} rubrique. On ne peut que sauter en avant et pas en arrière.

F - FICHIERS A ACCES DIRECT

F1 - Création d'un fichier ! CREATE

- * Syntaxe : !CREATE "(N°DISQ.)NOMFICH.FTYPE = NLU, LR, NE"
- * Paramètres : NLU = numéro logique qui sera affecté au fichier. C'est une valeur décimale, hexadécimale ou une variable. $1 \leq NLU \leq 255$.

FTYPE = D pour fichier à accès direct.

TYPE = le type du fichier sera mis d'office à .DAT, quel que-soit le type que vous avez indiqué.

LR = Largeur de l'enregistrement, compté en nombre d'octets. $LR \leq 255$

Une variable flottante occupe 17 octets

Une variable chaîne occupe 8 octets

Une variable chaîne occupe la longueur de la chaîne +2.

Exemple : Si C\$ = "TOTO", LA RUBRIQUE OCCUPE 4+2 c'est-à-dire 6 octets.

Si vous voulez écrire A, B%, C\$ dans un enregistrement, il doit avoir au minimum $17 + 8 + 6 = 31$ octets.

LR=31

NE = Nombre d'enregistrements réservés à l'avance

- * Action : Créer le fichier NOMFICH en accès direct composé de NE enregistrements de LR octets chacun, et lui associer le numéro logique NLU. C'est à vous de savoir ce que vous allez mettre dans chaque enregistrement pour calculer sa taille. Si vous avez réservé NE enregistrements, vous pouvez accéder directement à n'importe lequel de ceux-ci.

F-2 Ecriture ! WRITE

- * Syntaxe : ! WRITE NLU, E' < Liste de variables >
 ou ! WRITE NLU, E : REM <Liste de variables >
 pour la version ORIC 1
- * Paramètres : NLU = Numéro logique du fichier sur lequel on écrit.
 $1 \leq NLU \leq 255$, NLU = 0 interdit.

E= Numéro de l'enregistrement sur lequel il va écrire. E peut être une variable
 $1 \leq E \leq \#FFFF$

<Liste des variables> = Suite de variables séparées par des ",".
Tous les types de variables sont acceptés sauf la variable en tableau comme A\$(3,3).

Attention : Lorsqu'une variable chaîne est utilisée dans la liste de variables, elle doit exister déjà avant. Sinon il faut la définir par exemple par A\$ = " ", avant d'utiliser cette instruction.

- * Action : Ecrire dans le fichier à accès direct de numéro NLU, à l'enregistrement E, la suite des valeurs correspondant à la liste de variables. E doit être inférieur ou égal au nombre d'enregistrements réservés. E peut être supérieur de 1 au nombre d'enregistrements total. A ce moment on rajoute un enregistrement au fichier.

Seule la taille de l'enregistrement est fixe. Contrairement aux fichiers séquentiels, la taille des rubriques à l'intérieur d'un enregistrement peut être remodifiée car chaque fois c'est l'enregistrement entier qui est écrit. Vous ne pouvez pas écrire ou modifier séparément quelque rubrique particulière d'un enregistrement. La place restante d'un enregistrement est remplie par des "Espaces".

Attention : Avec l'ATMOS, les deux syntaxes sont valables
Avec l'ORIC 1, seule la syntaxe avec "REM" est conseillée

F-3 Lecture ITAKE

- * Syntaxe : ! TAXE NLU,E' <Liste de variables >
ou ! TAKE NLU, E : REM <Liste de variable> pour la version ORIC 1
- * Paramètres : NLU : Numéro logique du fichier sur lequel on lit.
 $1 \leq NLU \leq 255$ NLU= 0 interdit
E= Numéro de l'enregistrement sur lequel il va lire,
E peut être une variable $1 \leq E \leq \#FFFF$

<Liste de variable > = suite de variables séparées par des ",". Tous les types de variables sont acceptés sauf la variable chaîne en tableau comme A\$(3,3). Il ne doit donc pas avoir d'espace derrière le signe \$.

Attention : Lorsqu'une variable chaîne est utilisée dans la liste de variables, elle doit exister déjà avant. Sinon il faut la définir par exemple par A\$ = "" "", avant d'utiliser cette instruction.

- * Action : Lecture dans le fichier à accès direct de numéro NLU, de l'enregistrement N°E. E doit être inférieur ou égal au nombre d'enregistrements total du fichier. A chaque lecture, vous lisez depuis le début de l'enregistrement.

F-4 Indication du nombre total d'enregistrements : ! WHERE

Cette instruction s'emploie comme en fichier à accès séquentiel, voir chapitre E8.

G- OPERATIONS SUR LES MATRICES

G-1 sauvegarde de la matrice ou du tableau dans un fichier !MSAVE

- * Syntaxe : !MSAVE "(N°DISQ.) NOMFICH =NOMAT"
- * Paramètres : NOMFICH= Nom de fichier sur disquette.
TYPE= Le type de fichier sera d'office. ARY c'est-à-dire Matrice (ARRAY)
NOMAT= Nom de la variable tableau ou matrice du programme BASIC.
Si la matrice est A\$(5,5), NOMAT est A\$, il ne faut plus mettre les parenthèses.
- * Action : Sauvegarder la matrice de nom NOMAT du programme BASIC, sur le disque en lui attribuant le nom NOMFICH.AR.Y.

G-2 chargement de Matrice ou de tableau !MLOAD

- * Syntaxe : !MLOAD "(N°DISQ.) NOMFICH= NOMAT"

- * Paramètres : NOMFICH= Nom de fichier sur disquette
TYPE= Le type de fichier sera d'office. ARY c'est-à-dire (ARRAY)
NOMAT= Nom de la variable tableau ou matrice du programme BASIC. Si la matrice est A\$(5,5), NOMAT est A\$, il ne faut plus mettre les parenthèses.
- * Action : Charger le fichier de nom NOMFICH.ARY dans la matrice NOMAT dans le programme BASIC. Il va de soit que la matrice était définie et dimensionnée. Le système ne saura pas où se mettre la tête si vous chargez un fichier de matrice variable chaîne dans une matrice de variable réelle ou vice-versa. Il ne tient donc qu'à vous de bien gérer vos types de matrice, en mettant par exemple "S" dans le nom de fichier pour signaler une matrice chaîne. Si la dimension de la matrice à charger et celle de la matrice réceptrice ne sont pas identiques, vous ne retrouverez plus les valeurs au même endroit.

H- SAUVEGARDE DE L'ECRAN

H - 1 en basse résolution : !LSCR

- * Syntaxe : !LSCR "(N°DISQ.) NOMFICH"
- * Paramètres : NOMFICH= Nom de fichier sur disque
TYPE= Le type sera d'office. SCR pour SCREEN c'est-à-dire ECRAN
- * Action : Sauvegarder l'écran c'est-à-dire l'espace mémoire de #BB80 à #BFDF, sur disque en lui attribuant le nom NOMFICH.SCR

H - 2 en haute résolution : !HSCR

- * Syntaxe : !HSCR "(N°DISQ.) NOMFICH"
- * Paramètres : NOMFICH= Nom de fichier sur disque
TYPE= Le type sera d'office. SCR pour SCREEN c'est-à-dire ECRAN.
- * Action : Sauvegarder l'écran en haute résolution c'est-à-dire l'espace mémoire de #A000 à #BFDF sur disque en lui attribuant le nom NOMFICH.SCR.

I - EXECUTION AU DEMARRAGE :

I-1 Assignation pour le démarrage : !START

- * Syntaxe : !START "NOMFICH (.TYP)"
- * Paramètres : NOMFICH= Nom de fichier en 1 seul caractère alphabétique
TYPE= .BAS, .BIN, .CMD .BAS par défaut
- * Action : Prévenir le système que le programme NOMFICH.TYP sera lancé automatiquement après le chargement du système (BOOT). Avant de sortir d'un programme à lancement automatique, n'oubliez pas de remettre # CBED en # 1B pour ORIC 1 ou # CCB0 pour ATMOS. (Ex. : Doke # 1B, # CBED)

Exemple :

Inserer dès le début du programme l'instruction suivante :

Pour ATMOS : DOKE # 1B, # CCB0

Pour ORIC 1 : DOKE # 1B, # CBED

En langage machine, n'oubliez pas d'interdire les interruptions et ensuite de leur rendre la main.

Une fois le programme assigné, faites ! MASTER "NOM-SYS" pour figer la situation.

I - 2 Pas de démarrage automatique : !UNSTART

- * Syntaxe : !UNSTART
- * Action : Si vous ne voulez pas de démarrage automatique il suffit de faire ! UNSTART puis ! MASTER "NOMSYS".

ANNEXE 1.

OCCUPATION MEMOIRE

# 3F4	}	Registres d'entrée/sortie de la carte contrôleur
à # 3FF		
# 400	}	Lien entre les 16 K RAM overlay et la ROM BASIC
à # 4FF		
# C000	}	TDOS en RAM overlay
à # FFFF		

ANNEXE 2

COMMENT ENTRER DANS LA MEV PARALLELE

La particularité de l'ORIC et de l'ATMOS est d'avoir 16 K octets de mémoire vive (MEV) qui ont les mêmes adresses que la mémoire morte du BASIC, donc cachées par cette dernière et non utilisée. Nous l'appelons la MEV (ou RAM) "OVERLAY" ou parallèle.

C'est là dedans que se charge le T.DOS, vous laissant ainsi libre le reste de la mémoire. Pour des applications particulières, si vous voulez y accéder, c'est simple, la carte contrôleur gère pour vous. Il suffit de suivre le protocole suivant :

Pour entrer dans la RAM OVERLAY :

- 1 - Interdire toute interruption par SEI
- 2 - Mettre 127 dans # 30 E pour interdire les interruptions de la VIA
- 3 - Mettre 1 dans # 3FA, à ce moment tout accès aux mémoires de # C000 à # FFFF est dans la RAM OVERLAY

Remarque : Il faut interdire les interruptions avant d'entrer dans la RAM OVERLAY si les adresses du sous programme de traitement d'interruption ne sont pas mis à jour dans la RAM OVERLAY.

Pour remettre en situation normale avec la ROM BASIC :

- 1 - Mettre 0 dans # 3FA.
- 2 - Mettre 192 ou # C0 dans # 30E.
- 3 - Remettre les interruptions par CLI.

ANNEXE 3

ADRESSES DU CONTROLEUR

# 3F4	: COMMAND STATUS REGISTER
# 3F5	: TRACK REGISTER
# 3F6	: SECTOR REGISTER
# 3F7	: DATA REGISTER
# 3F8	: SIDE SELECT
# 3F9	: DCR DISK CONTROLER RESET
# 3FA	: ORMA OVERLAY RAM ACCESS
# 3FB	: ROMDIS
# 3FC	: DISK 1 SELECT
# 3FD	: DISK 2 SELECT
# 3FE	: DISK 3 SELECT
# 3FF	: DISK 4 SELECT

ANNEXE 4

TABLE DES MESSAGES D'ERREUR

Messages d'erreur	Code	Signification
EXITING FILE	1	Le fichier demandé existe sur disque
DRIVE NOT IN LINE	2	Vous avez demandé une opération sur une unité de lecteur qui n'a pas été défini au système par ! MOUNT
PROGRAM TOO LARGE	3	Le fichier traité est trop long. Vérifier vos adresses de début et fin.
FILE TYPE MISMATCH	4	Discordance de type de fichier.
DISK FULL	5	La disquette est pleine.
I/O ERROR	6	Erreur de lecture ou d'écriture. Vérifier les connexions. Vérifier si votre disquette n'est pas abîmée ou effacée. Si l'erreur persiste contactez le service après vente le plus proche.

FILE NO FOUND	7	Le fichier demandé n'est pas sur le disque.
WRITE PROTECTED	8	L'enregistrement, le fichier ou la disquette est protégé en écriture.
RANGE ERROR	9	Vérifier la taille de vos paramètres.
SYSTEM ERROR	10	Erreur grave de T.DOS.
SYNTAX ERROR	11	Vérifier la syntaxe et le type de variables.
TOO MANY OPENED FILES	12	Trop de fichiers ouverts en même temps. (16 maximum)
FREE MEMORY UNDERFLOW	13	Il ne reste plus de mémoire disponible. Voir HIMEM et FREE.
MISSING VARIABLE	14	Vous avez fait appel à une variable chaîne qui n'a pas été définie.
END OF FILE	15	Vous avez tenté une opération (lecture/écriture) en dehors du fichier existant.
OUT OF DATA	16	Vous avez tenté une opération (lecture ou écriture) en dehors de l'enregistrement en cours.

Le code correspond à la valeur que vous trouvez en # 489.

ANNEXE 5

EXEMPLES

Sur la disquette livrée avec l'appareil, vous trouverez des programmes d'exemples commentés.

TESTSEQU.BAS - Ouverture d'un fichier séquentiel.

SEQU2.BAS - Lecture et écriture sur fichier séquentiel à utiliser après TESTSEQU.BAS

TESTDIR.BAS - Exemples sur des fichiers à accès direct.

MATEST.BAS - Sauvegarde des Matrices.

GARANTIE

MATERIEL

Lecteur de Disquette "JASMIN" est garanti contre tous défauts de matière ou vice de fabrication pendant une durée maximum de six mois à dater de la date de facture, pour une utilisation normale de l'appareil. Le Numéro de série de l'appareil vendu doit figurer sur la Facture.

La Garantie ne s'applique pas :

- si l'appareil a été ouvert, a fait l'objet d'une modification, ou de réparation dans un atelier non agréé.
- s'il a été survolté, ou utilisé dans une atmosphère corrosive, ou dépassant 50 degrés
- s'il a été utilisé pour un usage différent de celui pour lequel il a été conçu
- pour les dégats causés par accident, incendie, chute ou introduction de liquide, ou à la suite d'un transport. (Le Transporteur est seul responsable).

Les réparations s'entendent pièces et main d'œuvre compris, le port toujours exclu, est à la charge de l'acheteur.

En cas de nécessité, le Fabricant se réserve le droit de remplacer totalement l'appareil pour un modèle plus récent, sans que l'acheteur puisse lui en faire le reproche.

LOGICIELS

Les logiciels, tel que TDOS, programme de démonstration, ou autres accompagnant l'appareil, sont mis à la disposition de la clientèle, et sont livrés gracieusement, Le Fabricant ne garantit pas les résultats des-dits Logiciels, et ne peut être tenu à aucune responsabilité quelconque directe ou indirecte suite à l'usage que peut en faire un utilisateur quelconque.

LES APPORTS DU NOUVEAU F-TDOS PAR RAPPORT AU T-DOS

Le F-TDOS pour FAST TDOS, hennis sa vitesse de chargement des informations 17 fois plus rapide que l'ancien TDOS, apporte à l'utilisateur 4 instructions nouvelles permettant d'avoir un accès direct aux secteurs.

!FS FREE SECTOR

Cette fonction permet de rechercher le premier secteur libre de la disquette

Paramètres à passer avant l'utilisation de la fonction !FS

Faire POKE#48C, No lecteur

Après l'exécution de !FS on trouve :

En #48D le No de la piste libre (NPL)

NPL compris entre 0 et 81 pour un lecteur double tête,

NPL compris entre 0 et 40 pour un lecteur simple tête

En #48E le No du secteur libre (NSL)

NSL compris entre 1 et 17

!WS WRITE SECTOR

Cette fonction a pour effet d'écrire un bloc de 256 octets de la mémoire centrale sur un secteur de la disquette déterminée par le No de lecteur, le No de piste, le No de secteur.

Paramètres à passer avant d'exécuter cette fonction :

#48C, No de lecteur

#48D, No de piste

#48E, No de secteur :

#48F et #490, adresse de début du bloc à transférer

Exemple :

Soit à transférer un bloc commençant en #3000 sur le lecteur 2, piste 18, secteur 5

POKE #48C,2

POKE #48D,18

POKE #48E,5

DOKE #48F,#3000

!WS

!RS READ SECTOR

Cette fonction permet de lire et de transférer en mémoire centrale un bloc de 256 octets, un secteur de la disquette déterminé par le No de lecteur, le No de piste, le No de secteur.

Paramètres à passer avant d'exécuter cette fonction :

#48C, No de lecteur

#48D, No de piste

#48E, No de secteur

#48F et #490, adresse de début du bloc à transférer

Exemple :

Soit à lire le secteur 5 de la piste 18 et le transférer en mémoire centrale à partir de l'adresse #3000.

POKE #48C,2

POKE #48D,18

POKE #48E,5

DOKE #48F,#3000

!RS

ATTENTION

Par !WS on peut écraser des informations déjà existantes sur la disquette donc il faut avant d'utiliser cette fonction, utiliser la fonction !FS afin de connaître la première piste / secteur libre.

!WS écrit un secteur puis le réserve s'il ne l'a pas encore été, afin que lors d'un !SAVE ou !CREATE etc... ce secteur ne soit pas écrasé. Ce qui a été écrit par !WS n'apparaît pas au catalogue.

!DS DELETE SECTOR

Cette fonction a pour effet d'effacer et d'enlever la réservation d'un secteur du disque.

Paramètres à passer avant l'utilisation de la fonction !DS

#48C, No de lecteur

#48D, No de piste

#48E, No de secteur

Exemple :

Soit à effacer un bloc sur le lecteur 2, piste 18, secteur 5

PC*E #48C,2
PC*E #48D,10
PC*E #48E,5
!DS

Par rapport au TDOS le F-TDOS permet de connaître lors d'un !LOAD*NOFICH.TYP* les adresses de début et de fin du fichier. Les adresses se trouvent en :

#48D et #48E : adresse de début
#48F et #490 : adresse de fin

!HELP

Intéressant lors de l'utilisation de la fonction !HELP le programme utilisateur en cours n'est pas perturbé, car il ne fait que recharger la page d'écran correspondant. Mais votre écran actuel est effacé.

Syntaxe :

!HELP*NOF*FONCTION*

Exemple : !HELP*CAT* vous donnera le mode d'emploi de cette fonction.

La nouvelle fonction est équivalente à !LOAD*NOFICH.SCR*

En plus de ces nouvelles instructions la disquette F-TDOS vous offrent un jeu d'arcade appelé par !"TTI" et un fichier à accès direct sur plusieurs clés appelé par !"REPEDITOR" qui vous servira pour votre fichier d'adresse, mais aussi comme exemple aléatoire et transformable à votre gré.

MANIPULATION SUR REPEDITOR

Après avoir lancé !"REPEDITOR" le menu suivant apparaît :

REPEDITORY-FICH

R--Recherche d'une fiche
A--Ajout d'une fiche
E--Effacer une fiche
S--Sauvegarde intermédiaire
F--Fin de session
L--Liste des fiches

TAPEZ VOTRE COMMANDE :

OPTION R

La recherche d'une fiche peut se faire :

NOM
CODE POSTAL
CODE CLIENT

Lors de la recherche par NOM si vous tapez une lettre suivie d'un astérisque vous obtenez toutes vos fiches à partir de cette lettre.

OPTION A

Cette commande vous permet de remplir votre fichier.

OPTION E

Cette commande permet de supprimer des fiches, pour cela vous devez rechercher la fiche à effacer par l'option R du menu, sélectionner l'option E du menu.

OPTION S

Cette commande vous permet de sauvegarder lorsque vous le désirez votre travail.

OPTION F

Cette commande vous permet de sortir de REPEDITOR.

OPTION L

Cette commande vous permet d'imprimer le contenu de votre fichier.

ATTENTION

Le nouveau F-TDOS est expédié avec les protections mécaniques, cette protection n'empêche pas d'utiliser la fonction !"EXP" pour le bon fonctionnement de REPEDITOR il faut déprotéger. Comme le nouveau F-TDOS auquel s'ajoute la fonction !HELP on ne peut plus utiliser le jeu et le fichier, il reste seulement pour un lecteur simple tête 44 secteurs disponibles. Si vous avez intérêt de prendre une disquette vierge et initialisée, de faire !MASTER (voir page 16) et pour le faire un backup complet de la disquette et détruire les autres fichiers.

Scanné par Andrec