

NOTICE d'UTILISATION

du

" RODOS SYSTEM "

I - INTRODUCTION

Ce système se compose de deux programmes : "RODOS" et "RECS" qui sont deux puissants jeux d'utilitaires combinés dans une seule ROM .

a /RODOS (shortcut pour ROM-DOS)

Ajoute à l'usage de l'Amstrad une nouvelle et complète dimension

avec :

- * Plus d'espace mémoire disponible pour l'utilisateur .
- * Lecture et Ecriture sur Diskette jusqu'à trois fois plus rapide.
- * Plus d'efficacité dans l'utilisation de l'espace mémoire des diskettes , donc moins de gaspillage de mémoire .
- * Les facilités d'un Silicon Disc sur 6128 (ou 464 / 664 muni d'une RAM compatible).
- * Création de Catalogues secondaires (Sub-Directories, comme avec le MS DOS des PC) pour organiser vos fichiers logiquement .
- * Redéfinition des Drives - (de A à H) .
- * Noms de Commandes du style UNIX / MS DOS .
- * Possibilité d'utilisation de Drives supplémentaires 5"1/4 , 3"1/2 , ou 3" - , en 40 ou 80 pistes simple ou double face .
- * Accès direct aux fichiers .
- * Spooler
- * Programme de Formatage résident (pas besoin de sortir son CPM du tiroir).
- * DUMP des Fichiers en HEXA , ASCII , et INFO sur les Fichiers .

b /RECS(ROM EXTENDED COMMAND SYSTEM)

Qui ajoute pour votre usage une foule de nouvelles commandes d'utilitaires .

- * Buffer d'imprimante (jusqu'à 64 K) .
- * Personnalisation de la Commande par barre 1 (commandes CP/M et RSX) .
- * Groupage des Bar-Commandes (Batch File comme sur PC avec MS DOS)
- * Interpréteur de lignes de Commandes
- * Changement de priorité de ROMs .
- * Invalidation d'autres ROMs (que celles du RODOS System) .
- * Lecture et Ecriture des secteurs sur d'autres disques (ex: Formatés MS DOS ou PC DOS)
- * Compatibilité avec mémoires additionnelles (comme les RAMs DK'Tronics 256 K)

II - RODOS (DOS en ROM)

RODOS est entièrement compatible avec l'AMSDOS . Les commandes normales du Basic comme " LOAD SAVE , OPENIN , etc) marchent avec RODOS . RODOS fonctionnera avec des disques formatés Amsdos

comme en utilisation normale (sans RODOS), RODOS est transparent pour l'utilisateur de disks Amsdos , bien que beaucoup de caractéristiques du RODOS, comme les " Directories " ne seront pas utilisables avec les disquettes formatées Amsdos .

Seul le formatage RODOS ouvre l'accès à toute une gamme de caractéristiques que l'on ne trouve normalement que sur des machines plus grosses et plus chères . Il s'y ajoute le Bonus d'un accès rapide et d'une utilisation plus efficace de l'espace mémoire des disquettes .

1 - Les "DIRECTORIES " (Ce sont des Catalogues structurés en arbre)

Pour ceux qui n'ont pas encore pratiqué les PC et le MS DOS , un "Directory" est l'équivalent du Catalogue Amsdos mais le système est en arbre à plusieurs étages avec des échelles verticales de communication comme entre les coursives et les ponts d'un navire . Sous Amsdos une Disquette peut être éclatée en " USER sections " qui sont en fait des " Directories " séparés mais à plat et sans communication entre elles. Ce système est assez peu maniable et impossible à structurer d'aucune façon . RODOS lui procure un Directory hiérarchisé , complètement structuré avec Directories et Sub-Directories (le pont principal en haut et les ponts inférieurs en dessous) Le Directory principal (le supérieur) est appelé " Racine " (Root) et un signe alphanumérique spécial le désigne : " / " .Ceci correspond au seul et unique Directory que l'on obtient quand on tape " Idir " sur un disque Amsdos . Avec RODOS on peut créer d'autres Directories , des "Sub-Directories "(sous catalogues) dans le directory racine , (ces sous-catalogues dépendent donc du catalogue principal) - Exemple :

Soit une diskette de jeux Basic contenant plusieurs genres de jeux . Quand on fait " Idir " ils apparaissent mélangés , soit :

```
SPELL RW  CYCLE RW      OXO RW      MAZE RW
KILLER RW SUNDOG RW  WHODUNIT RW  COLOSSAL RW
FIGHTER RW  PAWN RW
```

(RW = Read / Write)

On ne distingue pas leurs genres (aventure , stratégie , arcade) . Avec RODOS , la commande " MKDIR " (Makedir = Faire un Directory) permet de créer 3 Sub-Directories , soit : Arcades , Aventure , Strategie et de placer les jeux dans les sous-directories correspondants . En faisant : " IDIR " (sous RODOS) on obtient :

Aventure Dir, Strategie Dir, Arcade Dir

On peut alors maintenant utiliser la commande DIR d'une nouvelle façon pour regarder le contenu de ces sous-directories :

Faisons : IDIR,"/Strategie/" ceci fait apparaître OXO RW MAZE RW SPELL RW on peut également charger les fichiers en préfixant leur nom avec le chemin (Path) qui permet de les atteindre dans le sub-directory , soit :

IDIR,"/Strategie/OXO"

Si l'on veut regarder les titres dans un Directory , quel que soit son niveau , il suffit de installer comme Directory courant avec "CD " (Change Directory) ,

ex: ICD,"/Strategie" (ou ICD,"Strategie" sans "/" si "Strategie" est dans la racine) Quand le Directory ou l'on veut travailler (en LOAD ou en SAVE) est installé comme Directory courant il n'y a pas besoin du préfixe de chemin (le fameux PATH) .

Pour revenir au Directory racine (en haut , à l'air) , deux méthodes sont possibles :

!CD,"/" remplacera le Directory "en cours " directement à la racine (c'est l'ascenseur direct)

!CD,".." remplacera le Directory en cours sur celui directement au-dessus (c'est l'omnibus)

Nota : la recherche d'un sub-directory à partir de la racine se fait toujours vers le bas !... c'est un peu Shaddock mais c'est comme ça , c'est British et ça marche .(d'ailleurs , ne dit-on pas " manger les pissenlits par la racine " ?) .

2 - LES COMMANDES DU RODOS

La description des commandes comprend :

NAME - le nom de la commande
SYNOPSIS - la façon de l'utiliser avec ses paramètres
DESCRIPTION - ce qu'elle réalise
See Also - Voir aussi (d'autres commandes qui lui sont connexes)

La terminologie ci-après est utilisée dans la description de la commande :

FILENAME (Nom de Fichier) - Ce sont des noms comprenant jusqu'à 16 caractères . Si un point est utilisé le nom est éclaté en 12 caractères , un point , et 4 caractères d'extension.
PATH (Chemin) - Préfixe de sous-Directory indiquant le chemin pour aller au Fichier ou au sous-Directory recherché .
DIRECTORY (Catalogue) - Un nom de Directory a le même format qu'un nom de Fichier (Filename) .
WILDCARD (Joker) - Un nom de Fichier contenant un Joker ne s'appliquera qu'à la première entrée seulement (sauf spécification autre) . Tous les noms de Fichier peuvent comporter des Jokers (signes * et ?) à moins que le fichier ne soit à créer .

Nota : Lorsque les paramètres sont donnés sous la forme d'un mot , les possesseurs de CPC 464 doivent placer le mot dans une variable chaîne et passer cette variable chaîne au lieu du mot lui-même (à moins d'utiliser le " CLI " voir plus loin avec les commandes RECS le " Command Line Interpreter " .

ex : au lieu de !ERA,"Nom de Fichier"

il faut faire : A\$= " Nom de Fichier " puis !ERA,àA\$

A B C DRIVE

NAME DRIVE - Sélectionne le Drive courant par défaut .

A - Sélectionne le Drive A
B - - id - B
C - - id - C

SYNOPSIS !DRIVE,"drive-lettre" (la lettre du drive A , B ou ... etc)
!DRIVE,"drive-letter",drive-number (la lettre puis le numéro , o ou l ou n)
!A
!B
!C

DESCRIPTION Avec une lettre de Drive , minuscule ou majuscule , on désigne ce drive comme drive par défaut c'est à dire le drive de service (les autres étant allé au cinéma)
En ajoutant un numéro de Drive , on affecte la lettre logique à un drive physique , c'est à dire que le drive correspondant au nombre sera appelé par la lettre qui lui sera affecté .
ex : !DRIVE,"B",0 signifie que la lettre B est le nom du drive physique 0 , (le drive physique 0 (Zéro et non O comme oscar) est toujours le drive Floppy interne) - voir l'appendice A pour la liste des numéros de Drive et l'Appendice B pour la correspondance par défaut entre les lettres et les numéros de drive .

Les commandes !A , !B , !C , sont des versions abrégées de !DRIVE, "drive-letter" (lettres A , B etc) elles installent le drive courant sur un drive physique (bien se mettre dans la tête ces trois notions de : drive physique, drive logique et drive courant , elles sont énumérées ici dans leur ordre d'existence) , ainsi , la commande !A met le drive logique A - qui peut être le drive physique 0 ou 1 - en position de drive de service (courant comme disent les glissés) .

!A , !B , !C , ignorent tous paramètres donnés dans les Versions 2.15 et au-dessus .

BGET BPUT (Prendre Byte) (Mettre Byte)

NAME BGET - Prendre un octet d'un fichier ouvert et le mettre dans une variable (de type entier)

BPUT - Mettre un octet dans un Fichier ouvert

SYNOPSIS !BGET, à variable entière
!BPUT, Nombre ASCII d'un caractère Alphanumérique

DESCRIPTION

BGET: un octet est pris depuis le fichier et mis dans une variable entière donnée

ex : !BGET,aa% (La variable utilisée doit déjà exister et être entière . Si un nombre <0 est retourné , il y aura un message d'erreur tel que " End of File " .

BPUT: écrit le caractère donné dans le fichier.

ex : !BPUT,ASC"B" (asc"B"=66 en Décimal) mettra "B" dans le fichier.

voir aussi !POINT et !RANDOM

CD

NAME CD = Change Directory

SYNOPSIS !CD, "directory-name"

DESCRIPTION

Met en service comme directory d'utilisation celui portant le nom indiqué.

Le nom peut comprendre une liste de sub-directories séparés par un slash (/) constituant ainsi un chemin (Path) . Tous les noms doivent être des noms de Directories et non des noms de fichiers .

Ex : !CD, "/Games/Space" installe comme directory courant (en cours) le Sub-Directory " Space " lequel descend du sub-directory " games " lequel descend du Directory racine (tout en haut) (lequel doit sans doute descendre du singe comme nous tous) .

Les noms de Directory peuvent aussi être donnés de façon relative , c'est à dire à partir du Directory courant et non plus de la racine (c'est le cas dans nos familles , d'une part parceque le directory familial du haut est difficile à atteindre , d'autre part parceque on préfère se réserver la possibilité d'étendre un voile pudique sur une ascendance pas très flatteuse .

Mais quittons cette parenthèse philosophique pour retourner à nos racines (décidément c'est à la mode) et à cette notion de relativité du directory . En effet tout directory intermédiaire peut être considéré comme la racine du directory directement inférieur dépendant d'un nom de ce directory intermédiaire.

On peut donc descendre tous les étages les uns après les autres en les plaçant successivement en position de directory courant . Voilà pour la descente .

Dans le sens ascendant on utilise un nom spécial de directory ,soit ".." (deux points) qui signifie " Parent Directory " c'est à dire celui qui a engendré le Directory en cours . (La Parent Directory est en effet directement au-dessus dans l'arbre généalogique car il ne faut pas oublier qu'en anglais le mot "Parent" est le nom de l'ensemble Père+Mère ,sans la tante de Purgos . Donc , !CD,".." vous mettra au niveau du directory un cran au-dessus de celui auquel vous êtes . De proche en proche , on en arrive à l'étage du haut,la racine ,qu'on a intérêt à rejoindre par l'ascenseur direct en faisant CD,"/" si les étages intermédiaires ne vous intéressent pas . Si à l'étage au-dessus il y a directory du nom de "stuff" qui vous intéresse vous faites : !CD,"../Stuff" ce qui veut dire " remontez d'un cran jusqu'à l'étage des Parents et entrez dans "Stuff" .

Toute référence à un Fichier suppose que le fichier nommé est dans le directory courant(dans l'exemple précédent il faut que "Stuff" soit bien à l'étage). Si aucun nom de directory n'est donné , RODO5 opérera par défaut un changement de directory courant vers le dernier CD ayant intéressé l'étage (le directory ou sub-directory) .

Voir aussi !MKDIR et !DIR

CAT

NAME CAT - Catalogue la Diskette ou tout autre système contenant des Fichiers .

SYNOPSIS !CAT
!CAT,"File-system" (Diskette , Kasette , ROM , Hard Disk , CIA , DST , KGB etc...)

DESCRIPTION Affiche les noms des dossiers comme le fait la commande Basic " CAT " .
!CAT suivi d'un nom de système fera un CAT sur ce système :
ex : !CAT,"TAPE" donnera le catalogue de la Kasette

Voir aussi !DIR

COPY

NAME COPY - copie un fichier

SYNOPSIS !COPY,"Newfile","Oldfile" (Nouveau Fichier , Vieux Fichier)

DESCRIPTION Cette commande lit un Fichier et l'écrit dans un autre (en fait c'est une adaptation du PIP du CP/M qui marche en sens inverse du COPY de l'MS DOS
Ex : !COPY,"-TAPE-DOCUMENT.BAK","B:PAGES" copiera le fichier "PAGES" du Drive B: sur la Kasette sous le nom de "DOCUMENT.BAK" .Si un Joker est utilisé dans le nom de la source , tous les noms répondant au nom de fichier avec Joker seront copiés et le "Filename" sera ajouté à la fin du nouveau nom .

Ceci permet de faire de la copie d'un Drive à un autre .
Ex : !COPY,"B:","*.DOC" ou !COPY,"-TAPE-","*.Y"

Note : lors de copie de lots de fichiers ,il est recommandé d'utiliser l'option !opt,1,255 car elle annonce la liste des fichiers avant de les copier .De même , quand on utilise !COPY, RODO5 utilise la mémoire depuis \$8000 jusqu'à \$9000 comme Buffer (Tampon et non Tappeur) , tous les Data y existant sont bien entendu perdus .

DIR

NAME DIR - Affiche la liste du contenu d'un Directory

SYNOPSIS !DIR
!DIR,"Nom de Fichier"
!DIR,"Nom de Directory/Nom de Directory/Nom de Fichier"

DESCRIPTION De lui-même , seul , !DIR listera les Fichiers (et sous-directories) du Directory courant.
Avec les noms de fichiers optionnels il liste tous les fichiers répondant au nom spécifié . Le nom de fichier peut comprendre un chemin de directory (Directory Path) (voir CD) .
ex : !DIR,"letters/Jimmy.DOC" liste le fichier "Jimmy.doc" dans le sub-directory "letters" ,(s'il existe).
!DIR,"letters/Jimmy.*" liste tous les fichiers commençant par "Jimmy" dans les Directories du nom de "letters"
Avec un nom de Directory , !DIR liste les fichiers et les sub-directories dans ce Directory . Le dernier nom de Directory doit finir par "/" pour montrer qu'il s'agit d'un Directory et non d'un fichier .
ex : !DIR,"/games/adventures/" liste les fichiers du sous-directory "adventures" qui dépend du sub-directory "games" .

DISC ou DISK

NAME DISK - Sélectionne le système de traitement de Fichiers par Diskettes , en entrées et en sorties (Load et Save) en opposition à " TAPE " (Kasette) , etc ...
DISK.IN - Sélectionne le chargement seulement , la Sauvegarde restant inchangée .(Chargement par Disc ,sauvegarde par TAPE) .
DISK.OUT - Sélectionne la Sauvegarde sur DISC seulement

DISC = DISK
DISC.IN = DISK.IN
DISC.OUT = DISK.OUT

SYNOPSIS !DISK
!DISK.IN
!DISK.OUT

DESCRIPTION !DISK sélectionne le Disk comme support de Fichiers en entrées et en sorties .
!DISK.IN pour le chargement , le système précédent restant inchangé pour les sorties .

exemple: soit !TAPE (Entrées et Sorties sur KASSETTE)
puis !DISK.IN (Modifie l'entrée et la met sur DISK)
avec !DISK.OUT (c'est le contraire , l'entrée se fait par la KASSETTE et la sortie sur le DISK)

DUMP (Vidange de la mémoire)

NAME DUMP - Vide un Fichier en Hexadécimal sur l'écran (ou l'imprimante)

SYNOPSIS !DUMP,"Filename"
!DUMP
!DUMP,"Filename",Nombre (Adresse mémoire de départ du Dump)

DESCRIPTION Fait un Dump en Hexadécimal d'un Fichier donné sur l'écran . Quand un nombre est spécifié , ce nombre est l'adresse de départ du DUMP .
Pour une sortie sur imprimante on utilise la commande !PRINT(Voir plus loin à la section RECS)

EB

NAME EB - ERASE BACKUPS (Efface tous les Fichiers .BACK du Directory courant)

SYNOPSIS !EB

DESCRIPTION Tous les Fichiers du Directory courant avec extension ".BACK " sont effacés .
Equivalent à !ERA,"*.BACK"

Voir aussi : !ERA

ERA, RM,

NAME ERA - Efface un fichier (ou des fichiers s'il y a un Joker dans le Filename .
RM - kif-kif ERA

SYNOPSIS !ERA,"Filename" !RM,"Filename"
!ERA !RM

DESCRIPTION Le fichier spécifié est effacé (pas le Directory cependant) .
Le filename peut être précédé d'informations précisant le Directory .
Ex : !ERA,"Games/Game.DUF" efface le fichier "Game.DUF" dans le Directory "Games".
On doit Etre autorisé à écrire sur le fichier pour faire cela (Voir plus loin "Access")
Si l'on n'a pas donné de filename il vous en est demandé un .

Voir aussi !ERADIR, !DIR, !CD

ERADIR , RMDIR

NAME ERADIR - Efface le Directory
 RMDIR - Comme ERADIR
 !ERADIR,"Nom de Directory" !RMDIR," id "
 !ERADIR !RMDIR

DESCRIPTION Le Directory est annulé , MAIS S'IL EST VIDE SEULEMENT -.

Voir aussi !ERA , !CD , !DIR , !MKDIR

FORMAT

NAME FORMAT - Formate un Disk

SYNOPSIS !FORMAT,type - (type =le numéro d'un des quatres types définis ci-après)
 !FORMAT,type,Drive-number (Numéro de Drive)
 !FORMAT,type,Drive-number,Tracks (Nombre de pistes)
 !FORMAT,type,Drive-number,Tracks,Sides (Nombre de Faces)

DESCRIPTION Formate un Disk dans l'un des Format standards donnés par "type" :

0 - IBM (Version AMSTRAD)	2 - RODOS Format
1 - CP/M (-- id --)	3 - DATA (Version AMSTRAD)

L'option Numéro de drive précise le Drive physique ou faire le formatage .(Voir en Appendice les numéros de Drive) , s'il n'est pas précisé c'est le Drive par défaut qui est supposé choisi .

"Tracks" fait référence au nombre de piste sur le Disk (de 1 à 80)

"Sides" -----,id----- Faces du DISK (1 ou 2)

Quand on Formate le Silicon Disk , le nombre de pistes par défaut prend en compte toutes les pistes des Memory Banks inutilisées . Si l'on a besoin de moins de mémoire alors le nombre de pistes peut être calculé en tenant compte de ce que : chaque piste comprend 10 Secteurs de 512 octets (bytes) chacun ,autrement dit chaque piste mange 5 Kilo-octets .

NOTA Le Printer-Buffer doit être mis en place avant de formater le Silicon Disk si l'on doit s'en servir dans la RAM additionnelle de telle sorte que le Silicon Disk puisse prendre en compte toute la mémoire , jusque zé non compris le Printer Buffer , sans télescopage de mémoire .

TECHNICAL INFO Quand les types de Formatage 0,1 ou 3 , (cad les trois types normaux d'AMSDOS , excepté Format RODOS) , on ne doit utiliser que 40 pistes (numérotées de 0 à 39) . Le système d'exploitation AMSDOS suppose ce nombre et ne permet d'accéder qu'à une face : la face numéro 0 . Rodos , lui , permet n'importe quel nombre de pistes (de 1 à 255) et permet d'utiliser les deux faces d'une disquette (si le Drive a deux têtes) . Les quatre types de formatage utilisent des secteurs de 512 bytes , mais RODOS permet de 10 à 20 secteurs par piste (numérotés en hexa : &81 à &8A en simple face , et &81 à &74 en double face) , alors que les formats CP/M et DATA n'ont que 9 secteurs par piste (numérotés de &41 à &49 en Format CP/M , et de &C1 à &C9 en Format DATA . Le Format IBM , lui , n'en a que 8 (de &01 à &08) .
(Voir en Appendice C pour la disposition des renseignements sur le disque) .

Les Secteurs sont numérotés de &81 à &8A sur la face 1 du Disk et de &8B à &94 sur la face 2.
Les pistes sont numérotées de 0 à 39 sur un Disk de 40 pistes et de 0 à 79 sur un Disk de 80 pistes .

Notez SVP qu'avec RODOS un DISK peut être formaté à un nombre quelconque de pistes (1 à 255) et qu'il n'y a aucune vérification du nombre de pistes

Notez aussi que la plupart des lecteurs de Disquette permettent en réalité de mettre 2 ou 3 pistes de plus que spécifié , c'est à dire 42 pistes au lieu de 40 et 82 au lieu de 80 .

Exemples de Format RODOS :

!FORMAT,2,0 formate 40 pistes (0 à 39) sur une face (secteurs &81 à &8A)
soit 200 Ko

!FORMAT,2,1,82,2 formate 82 pistes (0 - 79) sur 2 faces (secteurs &81 à
&8A sur la face 1 , et &8B à &94 sur la face 2) , soit au total
820 K

INFO

NAME INFO - Information sur les Fichiers .

SYNOPSIS !INFO,"Filename"
!INFO
!INFO,"Filename",&t%,&S%,&L%,&x%

ou t% = type
S% = start (début)
L% = longueur (en nombre d'octets)
x% = adresse d'exécution

DESCRIPTION

donne des renseignements sur les fichiers sous la forme :

Name , type , load adress , + Size x adresse d'exécution

ou :

Name = Nom du Fichier
type = l'un de ces caractères spéciaux
- fichier Basic non protégé
% - fichier Basic protégé
! - fichier ASCII
& - fichier Binaire

Si aucun nom de fichier n'est spécifié tous les fichiers seront employés . Si le nom mis est suivi de quatre adresses variables , cela met les info concernant le fichier dans ces variables .

Voir aussi !DIR , !CAT , !DUMP , !LIST

LINK (Lien)

NAME LINK- crée une deuxième entrée dans un Directory en reliant cette entrée à l'entrée principale .

SYNOPSIS LINK, "Path/Filename"

DESCRIPTION Soit un nom de fichier qui n'existe pas dans le directory courant , cette commande créera une deuxième entrée vers ce fichier à partir du directory courant . A noter qu'il ne s'agit pas d'une autre copie du fichier en cause mais de ce même fichier à sa même place sur la disquette . Le lien "LINK" peut être réalisé même à partir d'un Drive différent, dans ce cas , le numéro du Drive (0 - 255) est stocké dans l'entrée ; pour utiliser le fichier , le disque d'origine doit être à la même place .
Effacer une entrée connexe créée avec LINK n'effacera pas le fichier d'origine .
Effacer le fichier d'origine invalidera tous les liens .

Attention - La nouvelle entrée tient lieu de fichier sur le disque , aussi , si ce dernier est effacé de sa position d'origine , ou sauvegardé ailleurs , alors l'information de position sera faussée et l'entrée connexe liée (linked) ne sera plus valable . C'est pourquoi il est recommandé d'utiliser "LINK" avec précaution .
Un bon emploi de cette commande consiste à configurer le Silicon Disk en Drive principal de travail et d'y créer comme un directory d'utilités relié à des fichiers résidant sur une disquette lue par un drive normal . De cette façon on met les utilités du Drive normal à la disposition du Silicon Disk (comme si l'on mettait un disque à l'intérieur d'un autre) . (Y a intérêt à faire la manip car c'est clair comme du jus de pipe de Skipper Ecossais) .
Voici donc un exemple qui va tout clarifier .
Faisons successivement :

MOD, "C:" (si C: est le Silicon Disk par exemple , cette commande lui donne la main)
ensuite LINK, "A:/utils/Assembler"
cette suite d'opérations crée une entrée du nom de " Assembler " sur le drive C qui permet depuis C: d'appeler le fichier " Assembler " qui fait partie du Directory "utils" lequel est sur la disquette qui est dans le Drive A:
Verstanden ? au fond c'est très simple , mais j'ai foutrement soif .

LIST

NAME LIST - Liste un fichier texte (sans commandes)

SYNOPSIS LIST, "filename"
LIST

DESCRIPTION Liste sur l'écran le fichier nommé . Si aucun nom n'est donné , le programme en demande un .
La sortie peut se faire sur imprimante via PRINT

Voir aussi INFO DUMP

LOAD

NAME LOAD - Charge un Fichier .

SYNOPSIS !LOAD,"Nom de fichier"
 !LOAD,"Nom de fichier",Bank
 !LOAD,"Nom de fichier",Start,Longueur

DESCRIPTION
Si l'on indique seulement un nom de fichier ,se dernier se charge à son adresse normale.
Avec une BANK , le fichier se charge à partir du bas de la mémoire de la Bank .
Avec adresse de départ et longueur , le fichier se charge à partir de l'adresse mémoire donnée par Start.

Voir aussi !SAVE

LS

NAME LS - Fait la liste des Fichiers (comme CAT et DIR)

SYNOPSIS !LS

DESCRIPTION Version style UNIX de DIR . Aucun fichier ou directory ne peut être précisé .

Voir aussi !DIR, !CAT

MKDIR MD (MD = shortcut pour MKDIR)

NAME MKDIR - Make Directory (Fait , Crée , un Directory)
 MD - comme MKDIR !

SYNOPSIS !MKDIR,"Nom de Directory"
 !MKDIR
 !MD,"Nom de Directory"
 !MD

DESCRIPTION Crée dans le Directory courant un sub-directory avec un nom donné .

ex: !MKDIR,"Project" crée le sous-directory "Project" dans le Directory courant .
On peut aussi créer un sub-directory plus bas que le directory courant , ainsi :

ex : !MKDIR,"Project/Old" crée un sub-directory "Old" dans le sub-directory "Project" qui est le directory courant .

NOTA : a - les deux commandes MKDIR et MD ne fonctionnent qu'avec les diskettes formatées RODOS .
b - si on ne donne pas de nom le programme en demande un .

Voir aussi !OD, et !ERADIR

POINT

NAME POINT - Déplace le FILE-Pointer

SYNOPSIS !POINT,position-number
!POINT,position-number,àvariable entière

Description

Déplace la position de lecture courante dans un fichier (et celle d'écriture dans le cas d'accès direct .

Si l'on fait : !POINT,1,aa%, aa% donnera la place courante dans le fichier , (on peut l'utiliser après !POINT,65535 pour obtenir la longueur d'un fichier).

```
ex : 10 OPENIN "Datafile"
      20 !POINT,20
      30 a% = 0 : GET,aa%
      40 PRINT "Le 20 ème octet du fichier est "a%;
```

Si le nombre indiquant la position de l'octet dépasse la fin du fichier, la position de lecture / écriture se déplacera à la fin du fichier . On peut utiliser cette procédure pour se déplacer à la fin d'un fichier , et , en utilisant la commande !RANDOM ajouter quelquechose à ce fichier .

```
ex : 10 OPENIN "Fichier de données" ( Data file )
      20 !POINT,65535 : REM ** Dépasse la Fin du Fichier **
      30 !RANDOM
      40 PRINT #9,"ajoute ce data"
      50 CLOSEIN
```

NOTA : ceci ne marche qu'avec les disques formatés RDDS .

Voir aussi !RANDOM !GET !PUT

RANDOM

(Littéralement " Au Hazard " et par extension " n'importe quand ", "n'importe ou " d'ou en informatique "Directement ") ,

NAME RANDOM - déclare un fichier d'accès direct

SYNOPSIS !RANDOM

DESCRIPTION

Ouvre un fichier d'entrées-permanent pour l'écriture . Toute sortie vers ce dossier s'écrit en surcharge sur le fichier existant (ou en addition à la fin) . CLOSEOUT retournera le fichier à un fichier d'entrée et écrira toutes les modifications sur le disque . CLOSEIN écrira toutes les modifs sur le disque et fermera le buffer (qui a servi de casserole pour faire cette cuisine) -

```
Ex : 10 OPENIN "Fichier de données "
      20 !RANDOM
      30 INPUT " Numéro d'enregistrement "; rec% ( rec = record )
      40 !POINT,rec% * 7
      50 INPUT #9,d$ :PRINT"Old Data";d$
      60 INPUT "New Data"; n$
      70 IF Len(n$) > 5 THEN 60 ( note discrète : LEN n'est pas un shortcut
      80 IF Len(n$) < 5 Then n$ = n$+ " ":GOTO 80 ( pour LENIN mais pour LENGTH - vezeihen sie
      90 !POINT,rec% * 7 ( alter kamerad .
     100 PRINT #9,n$
     110 GOTO 30
```


Cette routine permet de faire des modifs au milieu d'un fichier avec des entrées contenant jusqu'à 5 caractères (la longueur totale étant de 7 pour inclure CL et LF (Carriage Return et Line Feed).

Voir aussi !POINT !BGET !BPUT

REN

NAME REN - Renomme un fichier ou un groupe de fichiers

SYNOPSIS !REN,"New Filename","Old Filename"

DESCRIPTION

Renomme un fichier . Les chemins de directory peuvent être utilisés sur le "Old Filename " .
Les noms de fichiers avec Jokers (* et ?) s'appliqueront à tous les cas ad hoc rencontrés dans le directory courant .

Ex : !REN,"*.BAK","*.BIN" renommra tousles fichiers remplissant la condition et dans ce cas toutes les extensions .BIN seront changées en .BACK

NOTA Les Jokers ne peuvent être utilisés avec REN sur les disquettes formatées AMSDOS .

TITLE

(TITRE)

NAME TITLE

SYNOPSIS !TITLE,"Name"
!TITLE

DESCRIPTION

Cette commande donne un titre (ou une étiquette) à une disquette (RDDOS format seulement)
(si kif kif le MS DOS monzami , mizziane) .

Le nom peut avoir jusqu'à 16 caractères et comprendre des caractères de controle .

ex: !TITLE,"Games DISK" ou !TITLE, chr\$(12)+ "Fancy DIR " (efface l'écran et affiche un titre Fantaisie) .

Si aucun nom n'est donné le computer en demande un . Si le nom est "" alors le titre est configuré pour tous les espacements ...???(faut le faire pour voir , le glish est un peu radin d'explications , faut dire qu'il est écossais) .

Voir aussi !FORMAT

USER (en vranzé : utilisateur)

NAME USER - fixe le numéro d'utilisateur (cad la section USER du disque ou l'on va lire ou écrire)

SYNOPSIS !USER,number
!USER

DESCRIPTION kif-kif la commande CP/M (voir le manuel Amsdos à CP/M , on peut pas tout vous macher heap of BF slackers)

NOTA RODOS accepte des USER numbers de 0 à 65535 alors que l'AMSDOS n'en accepte que de 0 à 15, (sauf le fameux 229).
RODOS affiche tous les fichiers d'un Directory alors qu'Amsdos n'affiche que ceux du USER spécifié .
Aux numéros de USER entrés par !USER,number s'ajoute la valeur &FF00 , de telle sorte que l'on peut entrer seulement des nombres de 0 à 255 . (D'autres user ID's , (identificateurs) , sont réservés pour d'ultérieurs expansions et calculs) .
Le numéro de USER entré est ajouté logiquement à &0F et transmis au !USER CP/M
Remarque !USER tout nu = !USER,0 .

INSTALLATION D'UN SILICON DISK ET D'UN PRINTER-BUFFER

PRIMO - Bien lire les quatre sections suivantes :
FORMAT (dans la partie RODOS)
PRBUFF (dans la partie RECS)
APPENDIX A (Numéros des Drives physiques)
APPENDIX B (Détermination du Drive par défaut)

SECUNDO - y a un truc pour formater le silicon disk c'est qu'il faut mettre la disquette N° 1 du CP/M + dans le Drive A pour entrer le fichier système "SYS 128 "

TERTIO - Si l'on veut en même temps un Printer-Biffer et un Silicon Disk il faut absolument mettre en place d'abord le PrintBuffer sinon le Silicon Disk mangerait toute la mémoire

d'ou l'exemple de procédure suivant pour mettre en place un PRBUFF de 16 K et lez complément à 64 K en Silicon Disk :

```
faisons !PRBUFF ( tout seul pour avoir 16 k )
Ready répond le computer
faisons !C pour tater le Silicon Disk
DISC NOT FORMATTED répond le computer
```

A ce moment il faut mettre la Diskette du CP/M + dans le Drive A (sauf si elle y est déjà)

Puis on entre la formule magique:

```
!FORMAT,2,8 ( 2 c'est le numéro du Format Rodos )
( 8 -----"-----du Drive C ) Cfrr Appendix A et B
```

```
READY répond le computer
faisons !C pour voir si c'est bien OK
READY qu'y dit encore
```

faisons un petit
CAT pour voir K !CAT

```
Drive :C(8) )
) répond la machine au CAT
Bytes free = 044.5 k )
```

On fait la checksum : 16 K + 44.5 K = 60.5 K (il y a une petite perte de rendement qu'Einstein pourrait vous expliquer mais c'est normal) .

RECS (Rom Extended Command System)

Que vous utilisiez ou non RODOS , (et vous devriez !...) , vous avez à votre disposition un puissant jeu de commandes dans le Système " RECS " que nous allons exposer en détail maintenant.

Ses commandes variées étendront les moyens de contrôle du système d'exploitation à l'imprimante , l'écran , les RAMs et les ROMs , à l'exception de l'exploitation de l'homme par l'homme .

Ces commandes comprennent un Buffer d'imprimante , une personnalisation des commandes de ROMs , le changement de priorité des commandes individuelles , le tirage sur imprimante des écrans textes , la redirection des sorties d'écran sur imprimante ou disques , ou les deux !... (Fermez le Ban)

Plus jamais les possesseurs de CPC 464 n'auront à se battre pour affecter des variables chaînes à des commandes de ROMs , non ... plus jamais ça (on entend sangloter les barreaux de chaises) .

. SHIFT & FUNCTION - ENTER vous font entrer directement dans l'univers merveilleux du Superbe Interpréteur de Lignes de Commandes (c'est pas commercial ce laïus ? je suis mur pour une troisième carrière à la "CINO" et sans phosphate) .

Par exemple , sur le 464 (encore autorisé par la SPA) , renommer un fichier peut être la source d'un "Stress" parfois fatal avec ses vecteurs algorythmiques suivants :

n\$ = "nouveau nom "

o\$ = "ancien nom "

!REN,à\$,àø\$

Mais , avec le "CLI " , sorte de " ZORRO " informatique , (voir plus loin la manière de s'en servir) , tout devient rose il suffit de faire !CLI puis !REN,"nouveau nom","vieux nom"

N'avez-vous jamais eu ce problème ? :

```
10 - INPUT a$
20 - IF a$="CAT" THEN CAT
30 - IF a$="TAPE" THEN !TAPE
40 - IF a$="DISC" THEN !DISC
50 - IF a$="A" THEN !DRIVE,"A"
60 - etc .....
```

9999 GOTO 10

Pourquoi ne pas utiliser la commande !DO du RECS il suffit alors

d'écrire : 10 - INPUT a\$
20 - !DO,a\$

III . 1 RECS Commands

Les mêmes conventions que pour RODOS sont utilisées , soit :

NAME
SYNOPSIS
DESCRIPTION
FILENAME
PATH
WILDCARDS

(Voir le détail à RODOS 2.2)

Sont à noter également :

- SHIFT + CAPS LOCK qui verrouille CAPS LOCK
- SHIFT + CONTROL = PAUSE sur Scrolling de sortie sur écran .
- Si une disquette du nom de "DISC" se trouve dans les Drives "A" ou "B" , RECS va les lancer automatiquement à l'allumage ou au Reset (à moins que l'on presse la barre d'espace < SPACE > pendant le Reset ou qu'une touche quelconque soit pressée dans les premières secondes suivantes .

ACCESS

NAME ACCESS - Installer une protection de fichier

SYNOPSIS !ACCESS,"Filename", numero de Mode

DESCRIPTION

a / Pour Format RODOS seulement

Remplace les Modes de Lecture / Ecriture d'un fichier donné par un numéro de Mode qui est la somme des valeurs suivantes :

- | | |
|----------------------------|--|
| 1 - (Permission de lire) | 8 - (Permission Lire les autres USERS) |
| 2 - (----id----- écrire) | 16 - (----id---- Ecrire dans autres USERS) |
| 4 - (Exécuter seulement) | 32 - (Exécuter seulement autres USERS) |

Nota : 4 et 32 couvrent aussi 1 et 8 (évidemment)

Les autorisations de Lecture/Ecriture sont indiquées par "R" ou "W" après un " DIR " ou un " CAT " .

" X " indique une protection qui permet le chargement et l'exécution mais interdit " LIST " et " DUMP " .

Un programme Basic avec "P" après le filename ne peut qu'être exécuté .

Notez que vous devez être le seul détenteur et utilisateur du disque pour changer l'accès au fichier

Exemples d'application :

- !ACCESS,"myfile.BAS",27 autorise tous les utilisateurs à lire et à écrire dans le fichier (27 = 16+8+2+1)

- !ACCESS,"myfile*.!",45 autorise l'exécution seulement pour tous fichiers dont le nom commence par "myfile" .

b / Pour Format AMSDOS seulement

Change les Statuts " Read Only " et " Système " des fichiers .

0 - Fichier Normal

les numéros de Mode sont :

1 - SYS (Fichier Système - n'apparaît pas dans le Directory)

2 - R / O (Read Only - Ne peut donc être effacé)

3 - SYS + R/O

Voir aussi ! DIR

ALIAS

NAME ALIAS - Définit un "Pseudonyme" (Robert la terreur alias "Bob") pour une commande .

SYNOPSIS !ALIAS," Nouveau nom de Commande","Ancien nom "
!ALIAS

DESCRIPTION

Le nouveau nom de Commande est en fait transmis à l'ancien nom qui a toujours le pouvoir de déclencher l'exécution de la commande . Ceci permet de singulariser n'importe quelle commande à Barre en la nommant au gré de sa fantaisie (ex: la commande !Ma chérie rapplique ici dare- dare , peut être très heureusement "Aliasée" , hum .., par !LOVE) . Le glish qui lui , est moins poète vous propose les exemples suivants :

!ALIAS,"H","HELP,7" qui permet de faire !H au lieu de "!HELP,7"

et !ALIAS,"DEL","ERA TEMP" -----id----- !DEL au lieu de !ERA,"TEMP"

Note 1 : Si l'on fait !ALIAS sans paramètre la commande fera afficher tous les nouveaux noms de commandes avec leurs vrais noms et paramètres (ça vous rafraichit la mémoire) .

Note 2 : Le nouveau nom de Commande (le pseudo pour les anciens de réseaux) ne doit pas être le même que l'ancien . Si ya des têtus qui veulent absolument le faire , il leur faut utiliser les numéros de ROMs comme préfixe : soit !ALIAS,"DIR","7DIR#.BAS"

on peut se référer aux paramètres à1 à à9

en faisant : !ALIAS,"DIRECTORY","DIRa1 "

Mais RODOS ne vérifie pas ces paramètres .

ASKRAM

NAME ASKRAM - Demande la quantité de mémoire disponible.

SYNOPSIS !ASKRAM, à variable entière (ex : a%)
!ASKRAM, nombre, à variable entière

DESCRIPTION met le nombre de kilobytes disponibles dans la variable entière

ex: 10 a% = 0
20 !ASKRAM, a%
30 PRINT " Il y a : "; a%; " K disponibles "

Variante si ASKRAM est suivi d'un nombre , ex : !ASKRAM, 1, a% , a% va indiquer non pas un nombre de Kilo - octets , mais un nombre de Banks de 16 K

note: Il est à noter que le décompte de RAM ne porte que sur la RAM additionnelle et n'inclue pas les 64 K de la mémoire basse de base . Donc , normalement on doit avoir a% = 0 pour le 464 et le 664 et a% = 4 pour un 6128 (en effet , 4 x 16 = 64 (sauf en mathématiques modernes) c'est à dire les 64 K supplémentaires de la memory Bank du 6128) .

Voir aussi !PRBUFF et !FORMAT

CLS

NAME CLS - Clear Screen (Nettoie l'écran)

SYNOPSIS !CLS

DESCRIPTION Nettoie l'écran et le met au Mode 2 avec Texte en Blanc (INK 1,13) sur fond noir (INK 0,0)

CLI DO

NAME CLI - COMMAND LINE INTERPRETER (C'est transparent)

SYNOPSIS !CLI
!CLI, "Command String " (mot ou variable chaîne avec \$)
!DO comme !CLI

DESCRIPTION !CLI tout seul demande une commande barre qui pourra être entrée sans la barre .
Si on entre une ligne vide : !CLI, "" , on sort du CLI.
Tant qu'on est dans le CLI , les paramètres peuvent avoir soit une virgule , soit un espace , comme séparation (comme en CP/M) .
Les valeurs numériques peuvent être entrées en Hexa précédées de # ou & . Des chaînes sans espace peuvent être entrées sans guillemets (pourvu que le premier caractère ne soit ni une valeur numérique , ni un des signes # et & .
ex : 10 a\$ = "DIR "
20 INPUT b\$
30 c\$ = a\$ + b\$
40 !DO, a%\$

VOIR aussi : !HELP

EXEC

NAME EXEC - Exécute un fichier de commandes groupées .

Synopsis !EXEC,"filename"
!EXEC

DESCRIPTION Cette commande charge et exécute un fichier ASCII comme une série de Bar Commands. Toute ligne de fichier commençant par le caractère barre verticale (!) est envoyée au CLI (voir le CLI) . Les lignes sans barre au début sont envoyées à l'écran .
Ci-après , voici un exemple simple de fichier de commande de démarrage (kif-kif CONFIG.SYS sur PC)

```
!CLS
!PRBUFF          installe un Print Buffer de 16 K
!ALIAS,"H","HELP"  HELP alias H
!FORMAT,2,8       Configure un Silicon Disk
!DRIVE"A",8        nomme ce silicon disk "A" ( au lieu de C )
!A                prend A comme disque de travail
```

On peut évidemment imaginer ce que l'on veut mais il est bon d'avoir sur une diskette un petit fichier de ce genre pour avoir tout de suite votre configuration de travail au lieu de recommencer chaque fois l'appel de vos hommes (comptez, comptez les biens , y en a qui sont aux ...? et vous n'en savez rien) ça c'est seulement pour les moustachus qui ont été soldat .

NOTa Si quelque bar Command nécessite une entrée quelconque comme !LIST," ? " le programme EXEC s'arrêtera parcequ'il n'y a qu'un seul canal d'entrée de fichier lequel se serait fermé .

Il est utile de " !ALIAS " une command barre pour !EXEC un fichier particulier .

ex: !ALIAS,"S"."EXEC startup-file"

on peut également utiliser les paramètres à1 à à9 dans le fichier EXEC (voir ALIAS)

Voir aussi !CLI !ALIAS et surtout !FORMAT et !PRBUFF pour installer sans clash un Silicon-Disk et un Printer-Buffer .

FS

NAME FS - Filing System (Système d'enregistrement / Lecture des Fichiers)

SYNOPSIS !FS

DESCRIPTION Ajoute l'option de pouvoir utiliser un autre moyen que celui dans lequel on est installé tout en gardant ses pantoufles là où on est , exemple :

on est sur TAPE et on fait !TAPE

!FS

load "disc-game "

charge "game " depuis disc bien qu'en restant dans le filing système par cassette (TAPE)

Note : ceci est automatique avec !DISC ou !DISK mais pas avec !TAPE

Voir aussi !DISC et !DISK

HELP

Petit rébus avec CLI

ul ul

H

NAME HELP (os court)

H

Solution : Au s'cours , au s'cours, au s'cours
il manque " ul " dans un coin .

H

ul

SYNOPSIS

!HELP

!HELP,nombre

DESCRIPTION

!HELP tout seul fait la liste de toutes les ROMs ayant un numéro d'affectation dans le ROMBOARD (0 à 15)
!HELP,nombre (ce nombre étant un numéro de ROM) fera afficher sur l'écran la liste de toutes les
commandes à barre de la ROM correspondant à ce numéro .

Voir aussi

!ROMS et !ZAP

MOTOR ON

MOTOR OFF

NAME

MOTOR ON - Démarre le moteur du système Kasette
MOTOR OFF - Arrête ----- id -----

SYNOPSIS

!MOTOR ON

!MOTOR OFF

DESCRIPTION

Commande Marche / Arrêt du moteur du lecteur/enregistreur de kasette .

OPT

NAME

OPT - installe des options.

SYNOPSIS

!OPT,numéro d'options , n ou nn (valeur du paramètre de l'option)

DESCRIPTION

Cette puissante commande vous permet d'installer plusieurs configurations RODOS par défaut , soit :

- 1 , n - charger des messages (on / off)
- 2 , n - influence ou non des majuscules ou des minuscules . (on / off)
- 3 , n - messages de chargement des kassettes . (on / off)
- 4 , n - surcharge d'écriture (n = 0 , 1 , ou 2)
0 = demande Y / N , -1 = OK pour surcharger , 2 = OK mais faites un fichier BAK par prudence (ce
qui est interdit par la loi Amstrad et Liberté)
- 5 , n - compte des tentatives de lecture (16 par défaut) .
- 6 , n - messages d'erreurs des DISK (on / off)
- 7 , nn - avance moteur en 1/50 de sec (t= 1 sec par défaut)
- 8 , nn - retard moteur en 1/50 de sec (t= 7 sec -- id --)
- 9 , n - vitesse de piste des Drives en ms (t= 12 ms -- i d --)
- 10 , n - retard chargement lecture en ms (t= 1 ms -- id --)
- 11 , n - retard déchargt tête lecture en ms (t= 1 ms -- id --)
- 12 , n - nombre de connecteurs lecteurs disks extérieurs
- 13 , n - permet aux disques 40 pistes d'être lus sur Drives 80 piste en deux temps (on / off)
- 14 , n - temps d'installation tête lecture en ms (15 par défaut) .

Valeurs de n :

n = valeur d'un seul octet (0 à 255)

nn = valeur d'un double octet (0 à 65535)

off: n = 0

on : n = 255

PEEK POKE

NAME PEEK - Pique un octet dans la mémoire
 POKE - Fourre un octet dans la mémoire .

SYNOPSIS !PEEK,bank,address,àvariable entière
 !POKE,bank,address,variable entière (sans arobase)

DESCRIPTION "PEEK" lit le contenu d'une adresse mémoire définie par : Bank et address et met cette valeur dans la variable entière .
 "POKE" écrit une valeur de variable entière à une adresse définie par Bank et address
 Pour les deux commandes , les adresses à l'intérieur de la Bank sont comprise dans l'intervalle de 16 K (0000 - 3FFF).
 Ceci n'est valable que pour le 6128 ou les 464 / 664 avec mémoire supplémentaire

PRINT

NAME PRINT - redirection de l'affichage écran

SYNOPSIS !PRINT,n
 !PRINT

DESCRIPTION Met la sortie sur écran en accord avec les valeurs de n , n étant la somme des valeurs suivantes :

- + 1 - sortie écran vers fichier ouvert courant
- + 2 - sortie écran sur imprimante
- + 4 - pas d'affichage écran mais ça peut sortir ailleurs (si c'est par les narines appelez votre fournisseur habituel)

ces valeurs peuvent s'additionner dans n'importe quelle combinaison :

ex :

 !PRINT,1
 CAT Sauve une Hard copy du catalogue sur un fichier

 !PRINT,6 = 2 + 4 = sortie uniquement sur imprimante (entre nous je n'en vois pas tellement l'intérêt , sauf si le KGB bigle votre écran par derrière)

Voir aussi !PRBUFF !TDUMP !SPOOL

PRBUFF

NAME PRBUFF - Installe un Buffer d'imprimante

SYNOPSIS !PRBUFF
 !PRBUFF,bank
 !PRBUFF,Start,End
 !PRBUFF,Start,End,Bank

DESCRIPTION

Cette commande installe un Buffer (mémoire tampon) d'imprimante qui permet l'impression tout en libérant le computer pour d'autres taches .

!PRBUFF tout seul crée , par défaut , un Buffer de 16 K (une bank de 16 K)

NOTA TRES IMPORTANT

si l'on veut en même temps un Buffer d'imprimante et un Silicon Disk , il faut d'abord installer le Print Buffer autrement , lors du Formatage du Silicon Disk , ce dernier prendra toute la RAM pour lui tout seul , et le PRBUFFER se brossera .

Si l'on spécifie seulement un nombre (ex : !PRBUFF,3) le programme considère que ce nombre représente un numéro de Bank et la totalité des 16 K de la Bank est allouée au PRBUFFER .

Quand deux paramètres sont spécifiés , ils sont considérés comme les adresses de départ et de fin du Printer Buffer dans la RAM normale du computer (à l'usage des 464 et 664 qui n'ont pas de Bank)

Quand , enfin , trois paramètres sont spécifiés , les adresses de "Start" et "End" s'entendent comme étant à l'intérieur de la Bank spécifiée .

Nota : si la "End" adresse dépasse &3FFF (16383) on saute à-la Bank suivante .

Exemples : 1° - !PRBUFFER ,0,65535 installe un Printer Buffer de 64 K sur le 6128 (en prenant tous les 64 K des Banks) .

2° - Pour les 464 et 664 tout nus ,il faut logger le PRBUFFER dans l'espace mémoire de travail d'ou par exemple le petit programme suivant :

```
10 Memory 33000
20 !PRBUFF 33000,40000
```

Voir aussi : !PRINT !TDUMP !

READSECT

NAME READSECT - lit un secteur de Disquette physique

SYNOPSIS !READSECT,address,Drive number,piste,secteur

DESCRIPTION

Cette commande permet la lecture directe d'un secteur sur un disque physique d'un Drive donné et l'écrit en mémoire . ceci permet de lire des diskettes formatées sous d'autres systèmes d'exploitation comme MS DOS par exemple .A utiliser avec précaution cependant .

Précisions sur les paramètres :

address = adresse mémoire d'un buffer de 512 octets pour contenir le secteur .

Drive number = numéro de Drive physique (voir Appendix A)

Track = numéro de piste du disque (0 à 160)

Secteur = numéro de secteur du disque (+ 256 pour la face 2)

Voir aussi : !WRITESECT !OPT

ROMS

NAME ROMS - priorités des ROMs

SYNOPSIS !ROMS,rom-number 1,rom-number1...
!ROMS
!ROMS,rom-number 1,rom-number1....,"chaîne de démarrage"

DESCRIPTION Cette commande met en service les ROMs spécifiées . Les priorités normales sont rendues caduques par le nouvel ordre donné par la liste des numéros de ROM .
Toutes les ROMs installées sur le ROMBO et qui ne figurent pas sur la liste sont mises sur OFF .
!ROMS tout seul invalide toutes les ROMs à l'exception de celle du CP/M (s'il est sur la bonne chaussette, la n°7).

Note : !ROMS restaure le langage courant et tout le programme est perdu .
!ROMS avec une chaîne de démarrage fera s'afficher la chaîne de démarrage après le reset .
ex : !ROMS,2,3,"F"+chr\$(13) affichera " F " et fera RETURN (tout seul comme un grand) après avoir fait un reset de la machine et initialisé les ROMs 2 et 3 .

Note concernant les 464 RECS permet l'utilisation de 15 ROMs (de 0 à 14) . Quand le 464 est initialisé ,les ROMs 14 à 8 sont initialisées avant RODOS

VOIR aussi : !ZAP et !HELP

SAVE

NAME SAVE - Sauvegarde un block de mémoire .

SYNOPSIS !SAVE,"file-name",start,length
!SAVE,"file-name",Bank
!SAVE,"file-name",Bank,start,length

DESCRIPTION La première forme de commande comprend les paramètres suivants :
filename = nom du fichier à sauvegarder
start = adresse mémoire de départ (0 à 65535)
length = longueur ,c ta dire le nombre d'octets à sauver (0 à 65535)

La seconde forme spécifie le numéro de memory Bank de 16 K où l'on veut sauver le block . Ceci s'applique au cas du 6128 et des 464 et 664 avec RAM additionnelle .
Les valeurs des numéros de Bank utilisables sont les suivants :
(attention à ne pas se mélanger les pinceaux entre Bank de 16 K et bank de 64 K en lisant le tableau fait par le glish)

N° 16K BANK OU elles sont

- 0 - Bank 0 ,située dans la RAM de base de 64 k (de &4000 à &7FFF) des trois CPC
- 1 - Bank 1 ,---- id --- la deuxième tranche de 64 k du 6128 ou des 464 et 664 avec extra RAM de 64 K
- 2 - Bank 2 , à la suite de Bank 1
- 3 - Bank 3 , ----id----- 2
- 4 - bank 4 , ----id----- 3
- 5 - bank 5 et au delà (jusqu' à 14 dans le cas d'un 6128 avec une boîte de 256 k)

SPOOL

SPOOL

NAME SPOOL

SYNOPSIS !SPOOL,"filename"
!SPOOL

DESCRIPTION !SPOOL,"filename" ouvre un fichier avec un nom pour y sauver des données et en même temps envoyer ces données à l'imprimante .
!SPOOL tout seul clos le fichier et reset l'écran au mode normal .

Note : DIR et CAT ferment la sortie sur le fichier et ainsi SPOOL ne marchera pas avec ces commandes .Il faut utiliser !LS à la place de SPOOL pour imprimer CAT ou DIR (valable sous RODOS seulement) .

Voir aussi !PRINT

TDUMP

NAME TDUMP - envoie à l'imprimante un texte écrit sur l'écran .

SYNOPSIS !TDUMP

DESCRIPTION Envoie directement à l'imprimante le contenu d'un Texte écrit sur l'écran .Le non alphanumérique est traité comme des <SPACE> ,(ce n'est pas un dump graphique) .

Voir aussi : !PRBUFF !SPOOL

WRITESECT

NAME WRITESECT - écrit un secteur physique (sur une diskette)

SYNOPSIS !WRITESECT,address,numéro de Drive,piste,secteur

DESCRIPTION Cette puissante commande permet d'écrire directement dans le secteur physique des diskettes .Comme avec READSECT ,d'autres Formats de Disques peuvent être modifiés .

Les paramètres sont :

address = adresse mémoire d'un Buffer de 512 octets pour contenir les données à écrire .

drive-number = numéro de Drive Physique (Voir Appendix A)

track = numéro de piste physique (0 à 160)

sector = numéro de secteur physique (+ 256 pour la Face B)

Attention ! : cette commande peut détruire les informations et le Directory contenus dans le fichier ,et le rendre impropre à son usage normal .Donc , à utiliser avec précaution .

Voir aussi : !READSECT !OPT

ZAP

NAME ZAP - Invalide les ROMs

SYNOPSIS !ZAP,ROM number I,Rom number I,...
 !ZAP
 !ZAP,Eom-numbez,.....,"Startup"

DESCRIPTION

Cette commande met hors service les ROMs d'une liste donnée . Toutes les autres ROMs restent en service avec leur rang normal de priorité .

!ZAP tout seul met hors circuit toutes les ROMs

Note : !ZAP rétablit le langage courant

!ZAP avec "Startup" fera imprimer "Startup" au reset .

ex : !ZAP,3,"RUN"+chr\$(34)+"PROG"+chr\$(13) , cette commande invalidera la ROM 3 et démarrera un programme nommé "PROG"

Voir aussi !ROM et !HELP

APPEDIX A

NUMEROS DES DRIVES PHYSIQUES

- 0 : Normal Drive A
- 1 : Normal second Drive
- 5 : Normal DRIVE B (Double face)
- 8 : Silicon Disk (deuxième 64 K et +)
- 9 - 199 : Drivers externes supplémentaires
- 200 - 219 : Réseré aux Silicon Disks
- 220 - 254 : Réservés aux disques Durs
- 255 : Pas de Drive

APPENDIX B

DETERMINATION DU DRIVE PAR DEFAUT

Le système désigne les Drives par leur numéro qui est converti à partir de leur lettre ("A" à "H").
Ainsi , à toute lettre de Drive correspond un numéro de Drive et ainsi à un Drive réel (le physique).

Note : ceci ne s'applique pas aux programmes qui utilisent leur propre chargeur de fichier .

Drive letter - Drive réel

A	0	interne (664 et 6128)
B	9	deuxième Drive normal ou premier extérieur
C	8	Silicon Disk
D	10	deuxième Drive extérieur
E	11	troisième Drive extérieur
F	255	pas de drive
G	255	pas de drive
H	255	pas de drive

Nota : H 220 HARD DISK (version 2.15 et +)

Les nombres ci-dessus se trouvent dans une table à l'adresse &BE00 qui finit par l'octet (99) à &BE09 pour indiquer sa présence .

APPENDIX C

Aménagements des Disks et Renseignements sur les Secteurs

Tous les secteurs utilisés par RSDOS et AMSDOS sont de 512 octets.

Les renseignements suivants s'appliquent seulement au Format RSDOS. Les secteurs sont numérotés de &81 à &8A sur la face 1 des disques. S'il y a une face 2, ils sont numérotés de &8B à &94.

Un secteur peut stocker quatre différents types d'information :

- 1 - Blocks d'information sur le secteur racine .
- 2 - Directory
- 3 - Directory (suite)
- 4 - Fichiers

Le Block Secteur Racine c'est un secteur spécial qui contient les renseignements concernant le Disk (taille , etc...) . Ce secteur fait 512 octets de long ,dont les 256 derniers sont le Directory .Ce Directory est la racine qui est affichée lors d'un premier DIR sur le Disk .Ces 256 K sont composés comme suit :

<u>Octet N°</u>	<u>Renseignements donnés</u>
0	nombre de pistes -1 (cad numéro de piste les plus élevé).
1	nombre de faces -1 (0=simple face ,1= double face)
2 à 17	Nom du Disk en caractères ASCII
18 à 255	Disk Allocation Table
255 à 511	Root Directory (Directory racine)

Le Disk Allocation Block sert à déterminer quels secteurs du Disk sont utilisés , et ceux qui sont libres .
Ce Block est considéré comme une série de tableaux d'octets (commençant au Bit 0 de l'octet 0, puis continuant par les Bits 1,2,3,4,5,6,et 7 , suivis par le Bit 0 de l'octet 1 etc...)où chaque Bit représente un secteur du Disk . Le nombre d'octets utilisés dans ce Block dépend du nombre de pistes et du nombre de faces formatées sur le disk .

Un Bit de 1 indique que le secteur est utilisé .

Un Bit de 0 -----"----- libre pour utilisation .

Le premier Bit de ce Block est toujours mis étant donné qu'il représente le premier secteur du DISK (piste 0 , secteur &81) qui est le secteur racine lui-même . Les secteurs sont numérotés de &81 à &8A (et continuent de &8B à &94 sur un disque double face) . Les pistes sont numérotées à partir de 0 (zéro) .

La Disposition des Fichiers

Un fichier s'éclate en 508 tronçons d'octets lorsqu'ils sont stockés dans des secteurs ,avec 4 octets additionnels pour plus amples renseignements , (508 + 4 = 512)

Le secteur de 512 octets se décompose comme suit :

- 0 - piste des derniers 512 octets (ou piste Directory)
- 1 - secteur des derniers 512 octets (ou secteur Directory)
- * 2 - numéro de piste des 512 octets suivants
- * 3 - numéro de secteur des 512 octets suivants
- 4 - 511 octets du fichier

Note * ces octets contiennent la longueur des données du Block (cad leur nombre d'octets) si c'est le le dernier Block .

L'octet 2 = les huit bits low

L'octet 3 = le neuvième Bit de la longueur .

LA COMPOSITION D'UN DIRECTORY

Un Directory comporte trois sections qui sont :

- Le Directory Header (l'entête du Directory)
- Les Directory Entries (les entrées , cad les titres)
- Le End or Continuation Marker (l'indicateur de fin ou de suite) .

Le Directory Header il est long de 4 octets qui sont :

Octet N°	
0	- " D "
1	- " R " (ou " r " dans un secteur de suite)
2	- numéro de piste du Directory mère (père en anglais , c'est moins galant)
3	- numéro de secteur du Directory mère

Les octets 0 et 1 servent à vérifier que le Directory est vraiment un Directory et qu'il n'y a pas d'erreur .

Les octets 2 et 3 contiennent les numéros de piste et de secteur dans lesquels se trouve le Directory (ou les précédents piste et secteur dans le cas d'une suite .

Les Directory Entries sont chacune longues de 32 octets :

Octet N°	
0	- Octet d'accès au Directory
1 à 16	- Nom de fichier ou de directory
17	- octet d'accès au fichier
18	- piste de départ d'un fichier
19	- numéro de secteur de départ d'un fichier
20	- type de fichier
21 et 22	- longueur du fichier
23 et 24	- localisation
25 et 26	- adresse d'entrée
27 et 28	- numéro du proprio
29	- numéro du Drive lié (Linked)
30 à 31	- réservé pour future extension

L'Octet d'Accès au Directory

un '1' ou un '2' ont la valeur d'un marqueur de fin , tout autre nombre aurait la valeur d'octet de suite des bits significatifs , ainsi :

Bits N°

0 / 1	- marqueurs de fin d'octet
2	- s'il est mis il indique l'entrée dans un sub-directory .
3 à 5	- pas utilisés
6	- affiche le Flag du filename (0=le filename est affiché après un !DIR)
7	- existence du fichier (0 = fichier existe , 1 = effacé)

L'octet d'accès au Fichier

Il comporte aussi des bits significatifs et relève de deux groupes d'information.

<u>Bits N°</u>	<u>Action</u>
0, 1, 2	- rendent compte du possesseur du fichier (voir numéro du possesseur)
3, 4, 5	- rendent compte de tous les autres numéros d'utilisateurs (users)
0 et 3	- autorisation lecture (1 = read OK)
1 et 4	- ----"----- écrire (1 = Write OK)
2 et 5	- ----"----- exécuter (0 = execute)
6 à 7	- pas utilisés

Le Numéro de Propriétaire c'est un ID (Identificateur de 16 Bits)

Le Numéro du Link Drive il indique si l'on peut accéder au fichier par un Drive différent.
Les valeurs de l'octet et leur signification sont les suivantes :
254 = pas de LINK, 255 = le même disk, de 0 à 253 = nouveau numéro de Drive (voir !LINK)

Le END Marker Cet octet, indicateur de fin, suit la dernière entrée du Directory dans chaque
- secteur.

- Si cet octet est un 1, les deux octets suivants contiennent les numéros de piste et de secteur où se trouve la suite du Directory.
- Si cet octet est un 2, le Directory se termine là.

APPENDIX D

MEMOIRES TAMPON (Memory Buffer)

Tout fichier ouvert (en entrée / sortie / accès direct) déclenche la création dans la mémoire d'un enregistrement de tête. Cet enregistrement de tête est détaillé ci-dessous en même temps que les informations fournies en contrepartie du header par les routines d'ouverture du fichier du Firmware. (Ah ! c'est bien sûr clair comme de l'eau du Père Yé pollué par du Quoquaquola au triméthylbenzène) .

Octets

- 15 Type de Buffer (" D ", " I ", " R ", out / in / random
- 13 / 14 Longueur disponible dans le Buffer
- 12 numéro du Drive
- 11 piste du Directory
- 10 secteur ----"-----
- 9 et 8 longueur utilisée dans ce secteur
- 7 octet d'accès au fichier
- 5 et 6 Pointeur du Buffer
- 3 et 4 Quantité disponible pour lire dans le Buffer
- 2 numéro de piste courante
- 1 ----"--- du secteur courant

ce qu'on trouve dans le header :

Octets N°	
0 à 15	Nom du fichier
16 & 17	Pas utilisés
18	Tpe de fichier (basic , protégé ...etc...)
19 & 20	Longueur du fichier
21 & 22	adresse de début
23	not used
26 & 27	adresse d'entrée
28	piste précédente - départ du secteur tampon de 512 octets
29	secteur précédent
30	piste suivante
32 à 539	stockage des données du fichier

Les octets - 15 et - 13 d'une part et 28 à 539 sont ajoutés quand on accède au fichier la première fois , et entrent dans le buffer par DE pendant le Call du Firmware d'ouverture du fichier .

LE BUFFER D'IMPRIMANTE

Il fonctionne sur le principe d'une noria sans fin (tournant en rond) cad qu'une donnée est chargée à un bout et vidée en arrivant à l'autre faisant ainsi cheminer le Buffer à travers la mémoire .Quand le Buffer atteint sa limite haute il retourne vers sa limite basse et tourne ainsi en rond . Si le Buffer est en service ,RODOS vérifiera tous les 50 èmes de seconde s'il y a quelquechose en attente d'être envoyé à l'imprimante ,et s'il y en a , il l'enverra si l'imprimante est READY .
 Ce buffer peut être n'importe où en RAM, y inclus toute RAM d'extension adèquate , mais seulement jusqu'à une taille totale de 64 K . Tout caractère envoyé en &BD2B s'ajoutera dans le BUFFER .
 (PRBUFF appelle la routine MC WAIT PRINTER &BD2B pour voir si l'imprimante est en service et MC SEND PRINTER &BD31 pour envoyer le caractère) .

APPENDIX E

Les Numéros et Messages d'Erreur

Quand RODOOS annonce une erreur, il met un numéro d'erreur dans &BE6D de telle sorte que l'on peut vérifier quelle erreur (s'il y en a) a été annoncée. Un exemple de son utilisation pourrait être :

```
Poke &BECD,255    CLEAR ERROR
!MKDIR," newdir "
IF PEEK(&BE6D) = 13 THEN PRINT " newdir already existed " END
```

Les numéros d'erreur et leur message sont :

- 1 - TOO MANY PARAMETERS (Trop de Paramètres)
- 2 - BAD FILE NAME (Mauvais nom de Fichier)
- 3 - WRONG NUMBER OF PARAMETERS (Mauvais nombre de Paramètres)
- 4 - BAD DIR (Mauvais Directory)
- 5 - BAD CHARACTER (Pas le bon caractère)
- 6 - UNKNOWN COMMAND (Commande inconnue)
- 7 - ACCESS DENIED (Accès refusé)
- 8 - ACCESS NOT FOUND (Trouve pas accès)
- 9 - NO MATCH (Not found with wild card) = Rien qui colle
- 10 - DISK FULL (Disk plein)
- 11 - AMSDOS ? (Cannot find AMSDOS) = peut pas trouver l'AMSDOS
- 12 - CPM ROM MISSING (Checked during startup) = Manque CPM ROM (c'est vérifié au startup)
- 13 - DIR ALREADY EXISTS (Ce Directory existe déjà)
- 14 - BAD DRIVE (Tu t'as gouré de Drive et non pas le Drive est mauvais)
- 15 - UNKNOWN FILE SYSTEM (Système de traitement de fichier inconnu, ex : celui de la gendarmerie)
- 16 - INPUT FILE NOT OPEN (Fichier de chargement pas ouvert)
- 17 - OUTPUT FILE ALREADY OPEN (Fichier de sauvegarde déjà ouvert)
- 18 -REBORN (Ressuscité) (File not expected to exist)
- 19 -ALREADY EXISTS (Existe déjà)
- 20 - BAD FORMAT SPECIFIED (Pas bon format spécifié)
- 21 - CORRUPTED DISK ERROR (Disquette abîmée)
- 22 - DISK NOT FORMATTED
- 23 - BAD FILE
- 24 - DIRECTORY NOT EMPTY (Directory pas vide) ex : si on a fait !ERADIR et qu'y a encore du monde.
- 25 - CAN'T LINK TO A LINKED FILE (Peux pas joindre le fichier lié par !LINK)
- 26 - BAD ALIAS DEFINED (Mauvaise pseudo-commande)
- 27 - TOO MANY ALIASES (Trop de pseudos)
- 28 - DISC READ ERROR (Erreur de lecture)
- 29 - DISC WRITE ERROR (Erreur d'écriture)
- 30 - DISK TRACKING ERROR (Erreur de tracé de piste du disk)
- 31 - DISK MISSING (Disque absent)
- 32 - DISK FAULT (Erreur du Disk)
- 33 - DISK WRITE PROTECTED (Disk protégé contre l'écriture)

APPENDIX H

UTILISATION DE RODOS EN LANGAGE MACHINE

Pour utiliser les commande de RODOS en langage machine , il vous faut utiliser KL FIND COMMAND &BCD4 ,qui va chercher le nom de la routine et l'appeler , le registre A contenant le nombre de paramètres et le registre IX la liste des paramètres .

Notez que différentes versions de ROM ont des adresses d'entrée différentes , c'est pourquoi on ne peut tabler sur la constance de la valeur de ces adresses . Les CALL normaux aux adresses du FIRMWARE ,utilisés par RODOS fonctionnent sans modification et la plupart des programmes en code machine devraient fonctionner sans modification .

COMMAND INDEX

A	4	HELP	17	Appendix	
ACCESS	15	INFO	9		
ALIAS	15	LINK	10	A : DRIVE NUMBERS	22
ASKRAM	16	LIST	10	B : DRIVE SETTINGS (DEFAULT)	22
B	4	LOAD	11	C : DISC LAYOUT	23 - 24
BGET	5	LS	11	D : MEMORY BUFFERS	25
BPUT	5	MD	11	E : ERROR NUMBERS AND MESSAGES	26
C	5	MKDIR	11	F : PATCHING FOR HARD DISK USE	27
CAT	6	MOTOR.OFF	18	G : VARIABLE ROM WORKSPACE	27
CD	5	MOTOR.ON	18	H : USING RODOS FOR M/CODE	28
CLI	16	OPT	18		
COPY	6	PEEK	18		
DIR	6	POINT	12		
DISC	7	PRINT	19		
DISC.IN	7	PRBUFF	19		
DISC.OUT	7	RANDOM	12		
DISK	7	READSECT	20		
DISK.IN	7	REN	13		
DISK.OUT	7	RM	8		
DO	16	RMDIR	8		
DRIVE	4	ROMS	20		
DUMP	7	SAVE	20		
EB	7	SPOOL	21		
ERA	8	TDUMP	21		
ERADIR	8	TITLE	13		
EXEC	17	USER	13		
FORMAT	8-9	WRITESECT	21		
FS	17	ZAP	22		

*** ATTENTION DANGER ***

NE ROTISSEZ PAS VOS ROMS EN LES PLACANT A L'ENVERS SUR LES CHAUSSETTES (SOCKETS) . VERIFIEZ TROIS FOIS QUE LES ENCOCHES SONT BIEN AU NORD ET NON AU SUD .

(Pour les Myopes on a agrandi le dessin de la notice British).

REMEMBER - DAMAGE may be caused by inserting or removing ROMS with power on, OR inserting ROMS the wrong way round.



