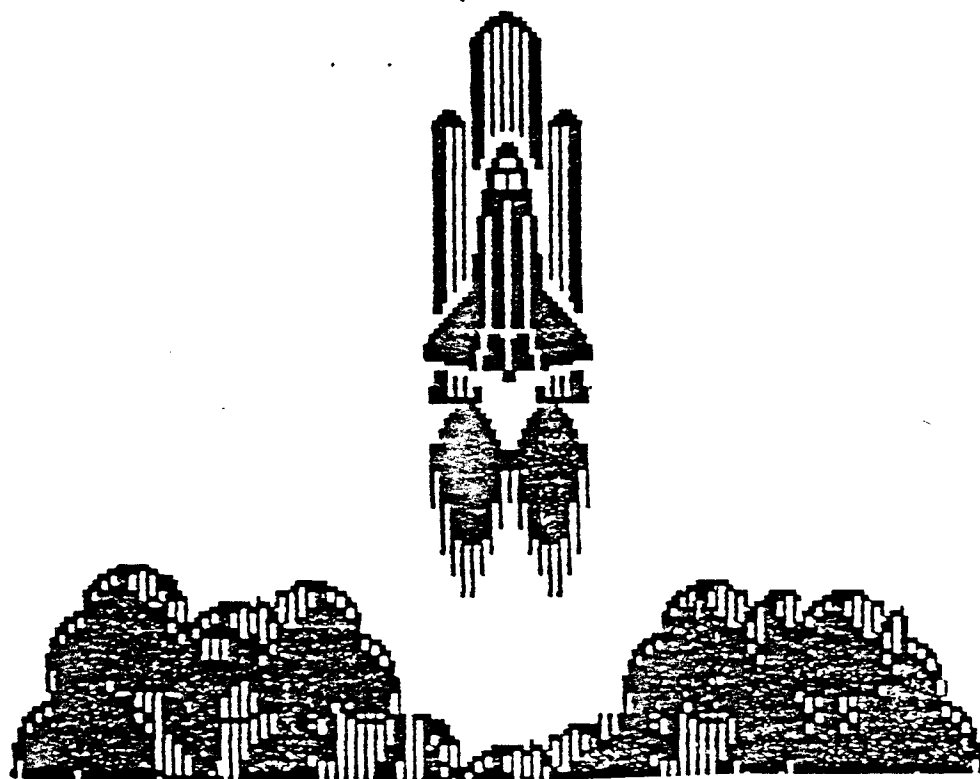


LA REVUE DU MICROCAM



RAMPE DE
LANCEMENT II

INITIATION A MSDOS

OCTOBRE 1987 N. 22A

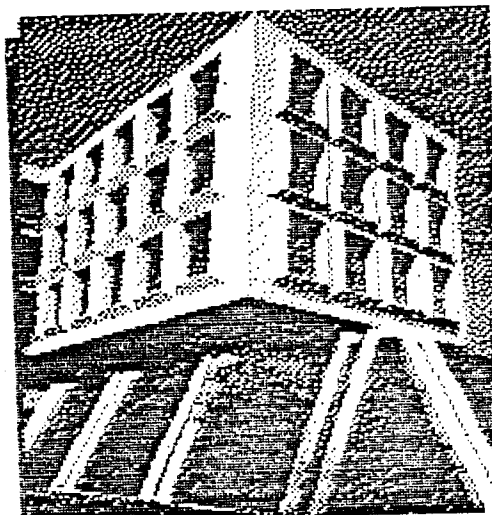
Réédition
Mai 1989

INITIATION A MSDOS

A MICROCAM, nous pratiquons la pluralité et la coexistence entre deux mondes APPLE et IBM.
Cette Rampe de Lancement concrétise une fois de plus cette pluralité.

Nous avons donné la "parole" à Lucien LOUCHE, un des membres fondateurs de MICROCAM et un des spécialistes du MSDOS au CREDIT AGRICOLE de l'Ille et Vilaine.

Y.R. CORNIL



UCA
MICROCAM

CREDIT
AGRICOLE
DES
ALPES
MARITIMES

LES AVENTURIERS DU BOUT DU MONDE

Numéro hors série RAMPE DE LANCEMENT

(c) Copyright LES AVENTURIERS DU BOUT DU MONDE
RENNES 1987

Dépot Légal: 641
ISSN 0295-9380

MICROCAM - CREDIT AGRICOLE
19 rue du Pré Perché 2025 X
35040 RENNES cedex

UCA MICROCAM 06
MICROCAM CREDIT AGRICOLE
111 AV. EMILE DESCHAMÉ
06700 ST LAURENT DU VAR

Directeur de la publication : Yves Roger CORNIL

Ce numéro a été réalisé par Lucien LOUCHE

Toute reproduction doit être soumise à notre autorisation préalable.

----- MS-DOS -----

- 1 - LE SYSTEME D'EXPLOITATION MS-DOS.
- 2 - ENVIRONNEMENT.
- 3 - LES COMMANDES ESSENTIELLES.

Le Système d'exploitation est un programme.

La plupart du temps écrit en langage machine pour des raisons de performances servant d'Interface logicielle entre l'ordinateur et son utilisateur.

Tout dialogue entre l'homme et la machine passe - directement ou non - par le système d'exploitation.

Lorsque l'ordinateur est équipé de mémoires de masse telles que disques magnétiques, le système d'exploitation prend alors le nom de DOS (Disk Operating System), ou SED en Français : (Système d'exploitation de Disquettes).

Qu'est-ce que MS-DOS ?

MS-DOS (de Microsoft DOS) est le nom du SED le plus répandu dans la catégorie des ordinateurs individuels construits autour du microprocesseur 8088 d'Intel ou équivalent (8086 et 80186/80188 du même constructeur). On parle surtout de lui depuis la commercialisation de L'IBM-PC en passe de devenir un standard.

Nécessité d'un système d'exploitation :

Devant l'importance grandissante du stockage offert par les disques ou les disquettes (de 160000 à 10 Millions de caractères et plus, selon le type,) l'utilisateur se rend rapidement compte de l'énormité du travail qui lui serait imputé s'il devait s'occuper lui-même de la gestion de l'espace disque et du dialogue avec l'environnement extérieur.

C'est pourquoi l'un des buts du système d'exploitation est de simplifier au maximum la tâche de l'utilisateur en prenant en charge la résolution de tous les problèmes ingrats et routiniers.

Fonctions remplies par un Système d'Exploitation.

a) - Un accès facile et rapide aux fichiers de données ou de programmes.

En pratique, il suffira de spécifier au SED le nom d'un fichier pour que celui-ci soit capable de l'extraire rapidement. Peu importe à l'utilisateur (en général ..) de connaître l'endroit exact où est physiquement implanté le fichier, (c'est le SED qui s'en charge), A l'inverse, le SED devra être capable de stocker (là où il y a de la place ...) le fichier que l'utilisateur lui confie, même s'il doit le morceler pour occuper les quelques "trous" qui existent sur la disquette.

b) - Le traitement des opérations classiques de maintenance des fichiers.

Il arrive fréquemment que l'Utilisateur éprouve le besoin de connaître les noms des fichiers disponibles sur le disque ou la disquette. Pour cela, il aimerait consulter le CATALOGUE ou REPERTOIRE.

Tel fichier n'est plus nécessaire : supprimons-le.
Tel autre est mal nommé : renommons-le. Ce fichier très important risque d'être détruit par mégarde : copions-le sur une disquette de sauvegarde.
Reste -t-il encore de la place sur cette disquette pour y placer ce petit programme ?

A toutes ces tâches, le Système d'exploitation est capable de répondre, moyennant la mise en oeuvre d'une simple commande.

c) - Un dialogue facile avec les organes périphériques (Entrée/Sortie).

Afin de résoudre les différences existant d'un organe périphérique à un autre, le SED contient des programmes spécifiques à ces périphériques qui se chargent d'intégrer toutes ces différences de comportement, en offrant à l'utilisateur des commandes évoluées permettant une banalisation des opérations d'Entrée/Sorties.
Là encore des commandes très simples permettent d'orienter des opérations spécifiques à tel ou tel périphérique. La seule précaution à prendre concernera le sens du transfert propre à chaque organe.

d) - Faire plusieurs chose à la fois (et bien...).

Enfin, certains SED comme ceux des gros ordinateurs, sont capables d'exécuter plusieurs tâches à la fois dans un contexte de (Multi-Tâches ou Multi-Traitement) ou de gérer plusieurs postes de travail (Multi-Poste).

En réalité, l'ordinateur exécute séquentiellement des "fragments" plus ou moins entrelacés de ces différentes tâches (sauf dans le cas où il est équipé de plusieurs Unités Centrales. On parlera alors de Multi-processeur.)

Ce n'est pas le cas des versions actuelles de MS-DOS. Toutefois, il est facile de remarquer que l'ordinateur semble parfois exécuter plusieurs choses à la fois, par exemple : Saisie de caractères au clavier durant une impression ou une opération sur le disque dur ou la disquette. C'est également le rôle du SED d'exploiter utilement les temps morts nécessaires aux opérations d'entrée/sortie. On parle dans ce cas de simultanéité des Entrées/Sorties, qu'il ne faut pas confondre avec les caractéristiques dont il est question plus haut.

/-----/

a) - CONFIGURATION

Un Micro-Ordinateur, pour pouvoir travailler, doit en outre être équipé d'une capacité mémoire de 64K octets au minimum pouvant atteindre 640K octets et même plus, d'une ou deux unités de disquette (ou d'un Disque dur : 10 Mo à 70 Mo dans le cas où la version MS-DOS le supporte.), une imprimante est également indispensable.

b) - FICHIERS et PERIPHERIQUES

Sous MS-DOS, un fichier stocké sur une disquette ou un disque est désigné par un nom composé de 1 à 8 caractères alphanumériques et spéciaux. Afin de faciliter leur exploitation, il est courant de classer les fichiers par Types :

Fichiers en langage Basic
 en langage Fortran
 en langage Cobol
 en langage Assembleur

Fichiers contenant des Textes, etc...

Le nom de fichier pourra donc être suivi d'un nom de type sur 1 à 3 caractères. On l'appelle l'EXTENSION du nom. Dans ce cas, le nom et l'extension devront être séparés par le caractère "."

Bien que l'extension soit facultative en général, elle devient obligatoire dans certains cas : Les commandes externes de MS-DOS par exemple, possèdent toutes l'extension "COM". De même, MS-DOS reconnaît implicitement quelques extensions spécifiques, comme notamment :

exe programmes exécutables issus d'une édition de liens.
bak fichiers de sauvegarde.
\$\$\$ fichiers temporaires
bas fichiers basic
bat fichiers "batch".

.../...

Les unités de minidisquettes (ou de disques durs) possèdent aussi un nom : il est composé simplement d'une lettre partant de A et suivie du caractère ":". Ainsi, "A:" désigne la première unité "B:" désigne la seconde, etc...

En plus des disquettes, certaines commandes de MS-DOS reconnaissent les noms de quelques unités "logiques" désignant des interfaces sur lesquelles peuvent être raccordés des organes périphériques :

CON: console formée d'un écran et d'un clavier.
LPT1: interface de sortie parallèle (imprimante).
PRN: idem LPT1.
COM1: interface série de communication.
AUX: idem COM1.
NUL: unité "fictive" généralement utilisée en simulation pour des applications de test.

c) - SYNTAXE UTILISEE DANS LES COMMANDES.

Les commandes MS-DOS obéissent à une syntaxe stricte mais relativement simple d'emploi. Elles peuvent être introduites en caractères Majuscules ou minuscules.

Les noms de fichiers ou de commandes sont formés de 1 à 8 caractères alphanumériques et ne doivent pas contenir d'espaces. De plus ils admettent les caractères spéciaux suivants :

\$ & @ # ! % ' () - < > _ \

Ces règles s'appliquent également aux extensions des noms qui sont formés de 1 à 3 caractères.

Tout autre symbole est exclu. Toutefois, les symboles "<", ">" et "\" ne doivent être utilisés que dans les noms de fichiers sous MS-DOS version 1 seulement.

Un nom de fichier est complètement défini en MS-DOS par la forme :

<nom unité de disquette><nom fichier>.<extension

Attention : Les signes "<" et ">" sont utilisés ici comme délimiteurs syntaxiques et ne doivent pas être inclus dans la définition.

EXEMPLE : B:nomfich.bas

désigne le fichier de nom : "nomfich" et d'extension : "bas" se trouvant sur l'unité : "B".

Au départ, MS-DOS considère toujours la disquette A comme unité implicite (C s'il s'agit d'un disque dur). Aussi, il n'est pas nécessaire de spécifier ce nom si le fichier appelé se trouve sur l'unité implicite.

Par contre, tout fichier se trouvant sur une autre unité devra avoir son nom précédé du nom de cette unité. Par exemple, si l'unité implicite est A et que l'on désire appeler le fichier "prog.txt" sur cette unité, sa référence sera : prog.txt
Si par contre ce fichier se trouve sur l'unité B, sa référence sera : B:prog.txt.

Pour changer d'unité implicite, il suffit de taper le nom de cette unité implicite suivi de la touche "RETURN"

Exemple : Nous sommes sur l'unité implicite : A:
et l'on désire rendre l'unité C (disque dur) implicite, il suffit de faire C: <Return>.

MS-DOS permet de manipuler non seulement des fichiers individuels ayant des noms explicites, mais aussi des groupes de fichiers dont les noms possèdent des similitudes, comme par exemple : tous les fichiers dont la seconde lettre du nom est C et dont l'extension est "BAS"... C'est la notion de référence ambiguë.

Lorsque un fichier est clairement défini (nom et extension formés de symboles alphanumériques selon les règles énoncées plus haut), on dit qu'il possède une référence explicite. Si par contre il comporte dans son nom ou son extension des symboles tels que "*" ou "?", nous dirons que sa référence est ambiguë.

Employé dans le nom ou l'extension d'un fichier, le symbole "?" signifie : "peut être remplacé par n'importe quel caractère" quant au symbole "*", il signifie : à partir de maintenant, peut être remplacé par n'importe quel caractère ou groupe de caractères".

EXEMPLES

A?BC peut désigner les noms ABBC, ACBC, A1BC, etc..
?XY peut désigner les noms AAXY, 1ZXY, XYXY, etc..
?ABC.??X peut désigner les noms XABC.TEX, 1ABC.HEX...
????.COM désigne tous les fichiers dont le nom a une longueur de 4 caractères et dont l'extension est "COM".
A*.* désigne tous les fichiers de l'unité implicite dont le nom commence par un A et dont l'extension est quelconque.
C:*. * désigne tous les fichiers de l'unité C.

d) - CODES DE CONTROLE.

Durant la frappe des commandes au clavier, certaines touches peuvent être utilisées pour remplir des fonctions particulières. Voici les principales :

<u>IBM-PC</u>	<u>STANDARD</u>	<u>FONCTIONS</u>
Ctrl Break	Ctrl C	Annule l'opération en cours.
Ctrl Entrée	Ctrl J	Continue l'entrée en cours sur la ligne suivante.
Ctrl Numlock	Ctrl S	Suspend l'affichage jusqu'à appui sur une autre touche.
Ctrl PrSc	Ctrl P	Mise en service recopie clavier sur l'imprimante jusqu'à nouvel appui sur ces touches.
Esc	Ctrl X	Annule la ligne en cours.
Shift PrtSc		Recopie le contenu de tout l'écran sur l'imprimante (Hard Copy).
←	Ctrl H	Efface le dernier caractère frappé.
⇐	Ctrl I	Tabulation.

L'heure est maintenant venue de "mettre sous tension" !

3 - LES COMMANDES ESSENTIELLES.

- a) - DATE
- b) - TIME
- c) - DISKCOPY
- d) - DISKCOMP
- e) - FORMAT
- f) - DIR
- g) - CHKDSK
- h) - RENAME
- i) - TYPE
- j) - COPY
- k) - COMP
- l) - DELETE (DEL, ERASE)
- m) - SYS
- n) - MGDH
- o) - Fichier BATCH
- p) - LES REPERTOIRES

a) - DATE

Consultation ou modification de la date.

A l'initialisation, le DOS affiche divers messages sur l'écran et peut demander la date et l'heure du jour.

Dans le cas où la date n'a pas été demandée par le système ou est erronée, il est possible de la modifier ou tout simplement de la consulter en frappant la commande : "DATE" :

```
A>date
Current date is Mon 1-01-1986
Enter new date:
```

ENTER

LUNDI

```
A>time
Current time is 11:01:31.24
Enter new time:
```

ENTER

A>

Nous voyons que le système affiche la date courante ainsi que le jour de la semaine (Monday = Lundi). Il attend ensuite l'introduction d'une nouvelle date. En frappant uniquement la touche Entrée, la date courante est conservée.

b) - TIME

Consultation ou modification de l'heure.

Le principe est le même pour l'heure qui affiche d'autre part les centièmes de seconde.

Mais on peut aussi "forcer" une date ou une heure sans passer nécessairement par une consultation préalable. Il suffit dans ce cas de frapper la commande "DATE" ou "TIME" suivie de la nouvelle date ou la nouvelle heure. :

```
A>time 12:00:00
A>time
Current time is 12:00:05.54
Enter new time:
```

ENTER

A>

L'exemple ci-dessus force l'heure courante à midi.

c) - **DISKCOPY**

Copie de disquettes.

Commençons par dupliquer une disquette :

Supposez que vous désiriez avoir une sauvegarde de la disquette contenant des fichiers auxquels vous tenez, appelons cela une sécurité.

Pour se faire vous utiliserez la commande "DISKCOPY".

Cet ordre n'est pas résident en mémoire comme c'est le cas pour d'autres commandes, mais sera appelé de l'unité implicite "C" (disque dur) qui contient toutes les commandes du DOS.

Il permet de créer le double d'une disquette. La commande "DISKCOPY" signifiera : recopier la disquette se trouvant dans l'unité "A" sur une autre disquette qui sera montée en temps voulu dans cette même unité "A".

Le déroulement de la copie sera le suivant :

A> DISKCOPY **ENTER**

Insert Source diskette in drive A:

ENTER

La disquette est alors copiée en mémoire...

Insert Target diskette in drive A:

ENTER

Le contenu de la disquette source, actuellement en mémoire, est recopié intégralement sur la diskette destination (Target).

Lors de la copie vous observerez les messages

suivants :

Copying 2 side(s)

Copy complete

Copy another (y/n)? n

POUR ARRETER

A>

Attention : La commande "diskcopy" détruira toutes les données éventuellement présentes sur la disquette destination.... Pensez aux conséquences dévastatrices que peut produire cette commande...

Syntaxe	Exemples
diskcopy diskcopy S: D:/1	diskcopy diskcopy a: diskcopy a: b: diskcopy a: b:/1
S : désigne l'unité Source, et D: le nom de l'unité destination. /1 : paramètre optionnel dans le cas d'une disquette simple face.	

d) - **DISKCOMP**

Comparaison de disquettes.

La disquette s'étant copiée sans incident, il sera bon de contrôler que la copie s'est bien réalisée, pour cela nous utiliserons la commande "DISKCOMP" (comparaison de disques). Elle s'exploite de manière similaire à la commande Diskcopy :

A)diskcomp **ENTER**
Insert first diskette in drive A:

La première disquette est alors copiée en mémoire...

Insert second diskette in drive A:

Le contenu de la première disquette est alors comparé au contenu de la seconde.

Lors de cette comparaison vous observerez les messages suivants. :

Comparing 2 side(s)
Diskette compare ok
Compare more diskettes (Y/N)? n
A)

Cette commande qui réside également sur disque, vérifie que deux disquettes supposées identiques à priori, le sont effectivement. Comme pour la commande Diskcopy, on peut ajouter le paramètre "/1" si l'on ne désire contrôler que la première face des deux disquettes. La commande affiche sur l'écran les coordonnées (piste/secteur) des zones non conformes.

Syntaxe	Exemples
diskcomp diskcomp S: D: /1	diskcomp diskcomp a: diskcomp a: b: diskcomp a: b:/1

S : désigne la première disquette, et D : la seconde dans le cas d'un système à deux lecteurs de disquettes. /1 : paramètre optionnel dans le cas d'une disquette simple face.

e) - **FORMAT**

Formatage de disquettes.

La commande "diskcopy" est principalement utilisée pour faire des copies conformes de disquettes, on dit aussi des backups. Celles-ci ne sont pas en général destinées à une utilisation courante mais sont conservées pour archivage.

Les disquettes, telles qu'on peut se les procurer dans le commerce, ne peuvent pas recevoir directement les fichiers ou les programmes utilisés sur votre ordinateur. Il faut leur faire subir préalablement une opération de "formatage". Cette dernière est effectuée automatiquement lors de la commande "diskcopy", dans le cas où la disquette n'est pas déjà formatée.

Le formatage consiste à écrire magnétiquement sur la disquette certaines informations invisibles à l'utilisateur mais qui sont pourtant nécessaires. On y trouve notamment les numéros de pistes et les numéros et la longueur des secteurs.

Sur l'IBM-PC, on utilise 40 pistes par face, chacune d'elle recevant 8 secteurs de 512 octets d'informations réelles. De plus, la version 2 de MS-DOS autorise un format encore plus dense avec 9 secteurs de 512 octets par piste.

Pour formater une disquette, on utilise la commande "FORMAT", en spécifiant le nom de l'unité sur laquelle on veut effectuer cette opération :

```
A) format a: ENTER
Insert new diskette for drive A:
and strike any key when ready ENTER
Formatting... Format complete

322560 bytes total disk space
322560 bytes available on disk

Format another (Y/N)? n POUR ARRETER
```

Attention : l'opération de formatage détruit toutes les données éventuellement présentes sur la disquette... Pensez aux conséquences dévastatrices que peut produire cette commande !...

Après formatage, deux valeurs exprimées en octets apparaissent à l'écran :
- La taille totale de la disquette.

- La taille disponible laissée à l'utilisateur.

Dans l'exemple ci-dessus, les deux tailles sont égales puisque la disquette ne contient alors aucun programme.

- Avant de formater la disquette, le programme contrôle le nombre de faces exploitables sur l'unité cible. Dans le cas où il s'agit d'une double face, la disquette sera formatée en double face.

Pour forcer le formatage sur une seule face sur une unité double face, on utilisera le paramètre "/1", comme pour les commandes "diskcopy" et "diskcomp" :

Nous venons de créer une disquette non-système prête à recevoir des fichiers ou programmes.

Toutefois, cette disquette ne pourra pas être utilisée lors de la mise sous tension de l'ordinateur car elle ne contient pas de système d'exploitation.

Pour créer une disquette système - et ne contenant à l'origine que le système -, il suffira d'ajouter le paramètre "/S" dans la commande "FORMAT" :

A)Format b:/s

Notez qu'il est possible de cumuler les deux paramètres :

A format b:/1/S ou A format b:/S/1

- Dans le cas où le nombre de pistes défectueuses vous semble trop important essayez un nouveau formatage de la disquette.

Syntaxe	Exemples
format format D: /S /1	format format a: format a:/s format a:/s/1

D: désigne l'unité sur laquelle l'on désire effectuer le formatage.
S indique que la disquette recevra le système après son formatage.
/1 Formatage de la disquette en "simple face".

f) - DIR

Listage des fichiers du répertoire

Afin de connaître la liste des programmes et fichiers figurant sur le disque dur ou sur une disquette... la commande "DIR", va nous permettre d'afficher celle-ci à l'écran

(DIRectoty = catalogue ou répertoire du disque ou de la disquette).

Exemple : c>dir
a>dir

```
DL      COM      32950   4/07/83   14:01
CLAVIER      138     1/01/80    0:02
CLAVIER1     138     1/01/80    0:02
NORMAL      174     5/09/86    9:34
NORMAL1     174     5/09/86    9:34
```

5 fichiers

1216512 octets disponibles

a)

Les cinq colonnes de la liste produite par la commande "DIR" sont respectivement pour chaque programme ou fichier : le nom, l'extension du nom, la taille occupée sur le disque en octets, la date et l'heure de création ou de mise à jour. La dernière ligne donne le nombre de fichiers visibles sur la liste.

Dans le cas où la liste ne peut être entièrement contenue dans l'écran, on peut spécifier le paramètre "/P" (pour pause) qui arrête temporairement le défilement de la liste lorsque l'écran est rempli. Il suffit d'appuyer sur une touche pour avoir l'écran suivant.

A>DIR /P

Il y a aussi une autre solution qui consiste à ne laisser afficher que les noms et extensions de fichiers. On utilise pour cela le paramètre "/w" (de Wide, affichage en largeur). La liste s'effectue alors à raison de six fichiers par ligne :

A>DIR /w

```

BASRUN  EXE    EDLIN  COM    DISKID  WTDATIM  COM
VIP-E   FEP      TRANSFS EXE    INSTAL  FONCTION
VIPIBM  EXE      CONF   EXE    FTF     NB
VIPIBMCL EXE     PARAM  EXE    FLIPFLOP BAT  FONCCL
PARAM                                     FONCNB  PC0448  TRAN    EMI
```

26 fichiers

1216512 octets disponibles

A>

On remarque que les noms ne suivent aucun classement particulier : ils apparaissent dans l'ordre où les fichiers ont été placés sur la disquette.

Durant l'exécution d'un "DIR", il est toujours possible de figer temporairement l'affichage en pressant sur les touches : CTRL et S (nouvel appui pour continuer.)

Si nous sommes sur l'unité implicite C: et que nous désirerions connaître le contenu de la disquette se trouvant dans le lecteur A: , il suffit de faire : DIR A:

La commande DIR peut aussi permettre de vérifier rapidement la présence ou l'absence d'un fichier ou d'un programme sur le disque ou la disquette.

Exemple : A>dir format.com

```
FORMAT COM 3816 5-07-86 12:00p
1 File(s)
```

La référence ambiguë peut aussi être utilisée pour isoler certains fichiers de la liste :

Exemple : A>dir d*.com

```
DISKCOPY COM 2008 5-07-86
DISKCOMP COM 1640 5-08-86
DEBUG COM 5999 5-07-86
3 File(s)
A>
```

Cu en combinant le paramètre /w :

```
A>dir ke*.* /w
KEYBUK COM KEYBIT COM KEYBSP COM
KEYBFR COM KEYBGR COM
```

Les exemples pourraient se multiplier encore, mais vous pouvez continuer se

Syntaxe	Exemples
DIR	DIR
DIR <nomfich>/p /w	DIR TEXTE
	DIR *.BAS
	DIR ?A*.B*/w
	DIR C:*.exe/w/p

<nomfich>représente la spécification partielle ou complète d'un fichier (nom d'unité, nom de fichier, extension), ambiguëou non.

Analyse disque ou disquette

Cette commande permet, après analyse de la table d'allocation et du répertoire, de donner un compte-rendu global sur l'état d'un disque ou d'une disquette. Il s'agit de CHKDSK (CHecK DiSK) qui signifie : vérification disque.

Exécutons la :

A>chkdsk

```

160256 bytes total disk space
  8704 bytes in 2 hidden files
120320 bytes in 23 user files
 31232 bytes available on disk

327680 bytes total memory
313632 bytes free

```

A>

Nous voyons qu'elle donne la taille totale de l'espace disque et celle de l'espace restant libre, mais aussi le nombre de fichiers visibles ("user") : ceux donnés par la commande DIR. Ceux donnés invisibles ("hidden") innaccessibles par la commande DIR et qui sont les deux fichiers du système d'exploitation.

Dans le cas où elles existent, les pistes endommagées et bloquées par la commande "format" apparaissent aussi dans cette liste.

Par ailleurs, la commande "chkdsk" est parfois capable d'effectuer certaines 'réparations' lorsqu'elle détecte des anomalies flagrantes, comme par exemple un fichier présent dans le répertoire mais absent sur le disque (dans ce dernier cas, le nom du fichier est retiré du répertoire). De ce fait, il est nécessaire que la disquette cible soit déprotégée en écriture au moment de l'exécution de cette commande.

syntaxe	exemples
chkdsk	chkdsk
chkdsk <s>:	chkdsk b:

h) - RENAME (REN)

Changement nom de fichier.

Les noms de fichiers, tels qu'ils apparaissent dans le répertoire, peuvent être changés si vous le désirez. Ainsi, la commande "chkdsk" qui n'est pas résidente dans le système d'exploitation pourra être rebaptisée : "verif.com"... On utilise pour cela la commande "RENAME" ou "REN" en abrégé :

```
A>rename chkdsk.com verif.com
A>
```

Pour vérifier que cette instruction s'est bien exécutée, faisons :

```
A>dir chkdsk.*
File not found
A>
```

"chkdsk.com" ne figure effectivement plus dans le répertoire. qu'en est-il de "verif.com" ?

```
A>dir v*.*
VERIF    COM      1720    5-07-86    12:00p
A>
```

Cette commande est bien là ! et elle fonctionne... :

```
A>verif
160256 bytes total disk space
 8704 bytes in 2 hidden files
120320 bytes in 23 user files
 31232 bytes available on disk

327680 bytes total memory
313632 bytes free
A>
```

syntaxe	exemples
rename < ancien > < nouveau > ren < ancien > < nouveau >	rename fich fichier ren A:texte.bas fich.* ren text.bas *.doc

Commande résidente

i) - TYPE

Affichage fichier ASCII

"TYPE" est une commande résidente du système d'exploitation qui permet d'afficher sur l'écran le contenu d'un fichier, à condition naturellement que celui-ci soit lisible. Le fichier "autoexec.bat" par exemple, est uniquement composé de caractères affichables :

```
A>type autoexec.bat
keybfr
wdatim
REM The IBM personal Computer DOS
REM Version 2.11 (C) Copyright IBM Corp 1984,1985
A>
```

Essayez avec un fichier binaire :

```
A> type SYS.COM
8"1T0118070'0000|f1--- -!7+
```

... et ce n'est là que le début des injures !...

Syntaxe	Exemples
type < nomfich >	type prog.doc type A*.sou type A:texte.*

Commande résidente

j) - COPY

Copie de fichier - Concaténation

La commande que nous allons étudier, contrairement à la commande "diskcopy", permet de recopier un ou plusieurs fichiers ou programmes.

```
A>copy chkdisk.com  chk.com
      1 file(s) copied
```

A>

Dans cet exemple, le programme "chkdisk.com" (encore lui !) a maintenant un frère jumeau : "chk.com". Les deux programmes existent maintenant sur la disquette. Ils portent des noms différents (le système d'exploitation n'accepterait pas qu'il en soit autrement), et représentent tous deux le même programme : même longueur, même date de création :

```
A>dir chk*.*
CHKDSK  COM      1720  5-07-86      12:00p
CHK      COM      1720  5-07-86      12:00p
      2 file(s)
```

A>

La syntaxe générale de la commande "copy" est la suivante :

COPY <source> <destination>désignant, la plupart du temps, une spécification de fichier (unité de disquette + nom de fichier). Mais ils peuvent aussi désigner une unité logique :

< source > CON: (clavier de la console)
 COM1: ou AUX: (entrée ligne de communication)

< destination >
 con: (écran de la console)
 com1: ou aux: (sortie ligne de communication)
 lpt1: ou prn: (imprimante parallèle)

Ainsi, rien ne vous empêche de copier sur la disquette un fichier entré par l'intermédiaire du clavier (CON:) :

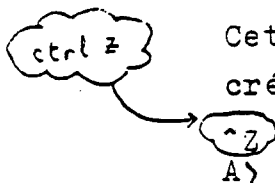
```
A>Copy con:texte.txt
```

Ceci est un texte composé à partir de la commande "copy con: texte.txt"

Cette commande peut donc être utilisée pour créer des petits programmes ASCII sur disquette.

```
1 file(s) copied
```

A>



Ce fichier dont le nom "texte.txt" (par exemple),
devra se terminer par un caractère spécial indiquant la fin (marque fin de
fichier ou EOF). Ce caractère sera obtenu par un appui sur les touches :
CTRL et Z (ou touche F6 sur l'IBM PC).

On peut afficher son contenu sur l'écran :

```
A>TYPE TEXTE.TXT  
-----
```

Il existe une autre façon d'afficher le contenu
d'un fichier sur l'écran...

Essayez la commande suivante :

```
A>copy autoexec.bat con:
```

```
keybfr
```

```
wdatim
```

```
REM The IBM Personal Computer DOS
```

```
REM Version 2.11 (C) Copyright IBM Corp 1984,1985
```

```
1 File(s) copied
```

```
A>
```

De la même manière, nous pourrions envoyer le contenu d'un fichier (attentio
pas n'importe lequel !) sur l'imprimante :

```
A>copy autoexec.bat LPT1:
```

```
1 File(s) copied
```

```
A>
```

Naturellement, la commande "copy" permet de copier des fichiers entre
différentes unités de disques ou disquettes :

```
A>copy chkdsk.com C:
```

```
1 File(s) copied
```

```
A>
```

Nous aurions aussi bien pu écrire :

```
copy chkdsk.com C:chkdsk.com
```

... mais si vous souhaitez conserver le même nom, seul le nom de l'unité
cible est suffisant.

Nous pouvons aussi profiter de la copie pour change
le nom du fichier destination :

```
A>copy chkdsk.com C:verif.com
```

```
1 File(s) copied
```

```
A>
```

Nous avons maintenant deux fichiers sur le disque

dur C:

A>dir C:

```
CHKDSK COM 1720 5-07-86 12:00p
VERIF COM 1720 5-07-86 12:00p
2 File(s)
```

Pour exécuter 3verif.com par exemple (qui n'est autre que "chkdsk.com"..), il suffit de faire

A>verif

Bad command or File name

A>

pour s'apercevoir que ça ne marche pas !

En effet, n'ayant pas précisé le nom de l'unité sur laquelle se trouve ce fichier, le système d'exploitation le recherche sur l'unité implicite qui est "a:"... et ne le trouve pas. Pour exécuter "verif.com", il faudra faire précéder son nom du numero de l'unité sur laquelle il se trouve :

A>C:verif

On peut changer facilement le nom de l'unité implicite (ou active), en frappant celui-ci, tout simplement :

A>C:

C>

C : est l'unité active

Maintenant, nous pouvons tester l'unité C avec la commande "verif" en frappant :

C>verif

Reprenons l'unité A comme implicite...

C>A:

A>

Découvrons maintenant comment la commande "copy" peut également dupliquer d'un seul coup plusieurs fichiers obéissant à un critère donné.

A>copy *.exe C:

EXE2BIN EXE

LINK EXE

2 File(s) copied

A>

Dans cet exemple, tous les fichiers de l'unité A dont l'extension est "exe" seront copiés sur l'unité "C" où ils conserveront le même nom. Remarquez que la commande "copy" affiche sur l'écran le nom des fichiers au fur et à mesure où ils sont copiés.


```

A>dir C:
CHKDSK  COM    1720  5-07-86  12:00p
VERIF   COM    1720  5-07-86  12:00p
EXE2BIN EXE    1280  5-07-86  12:00p
LINK    EXE   41856  5-07-86  12:00p
      4 File(s)

```

A>

Notre disque dur contient maintenant 4 fichiers.

Toutes les astuces à base de références ambiguës peuvent naturellement employées. Dans l'exemple qui suit, les fichiers de l'unité "A" ayant l'extension "exe" sont recopiés sur l'unité "C" mais en remplaçant au passage les deuxième et troisième lettres de leur nom par "aa

```

A>copy *.exe C:?AA*.*
EXE2BIN  EXE
LINK     EXE
      2 File(s) copied

```

et notre disque C: contient maintenant 6 fichiers :

```

A>dir C:
CHKDSK  COM    1720  5-07-86  12:00p
VERIF   COM    1720  5-07-86  12:00p
EXE2BIN EXE    1280  5-07-86  12:00p
LINK    EXE   41856  5-07-86  12:00p
EAA2BIN EXE    1280  5-07-86  12:00p
LAAK    EXE   41856  5-07-86  12:00p
      6 File(s)

```

A>

Une question peut toutefois se poser : peut-on être sûr de la validité d'une copie ? Une erreur peut parfois se produire lors de la recopie bien que cela soit peu fréquent.

La commande "COPY" nous offre un moyen qui permet de ne plus avoir de doute à ce sujet. Il s'agit du paramètre "/V" (pour vérification). Son utilisation est simple, il suffit d'ajouter ces deux caractères à la fin de la commande :

```

A>copy chkdisk.com c:/v
      1 File(s) copied

```

A>

La présence de cette option force le système d'exploitation à relire le fichier après écriture, afin de vérifier qu'il a été correctement recopié.

La commande "COPY" possède deux autres paramètres : "/a" et "/b". Ils servent à préciser si la copie porte respectivement sur des fichiers au format Ascii ou Binaire. Dans le premier cas, la copie se termine par l'écriture d'une marque fin de fichier (EOF) qui correspond au caractère Ctrl Z. Cette marque n'est évidemment pas écrite dans le second cas. Si aucun de ces paramètres n'est précisé, "/b" est pris par défaut.

Autre possibilité de "copy" : la fusion de fichiers Ascii (concaténation). Elle consiste à copier en séquence dans un unique fichier cible, différents fichiers dont les noms sont séparés par le signe "+". Soit par exemple, un second fichier texte comme celui-ci :

```
A>copy con: texte2.txt
Seconde partie du texte
à fusionner avec "texte.txt"
^Z
1 File(s) copied
```

A>

Nous voulons maintenant l'ajouter "derrière" le fichier "texte.txt" créé précédemment, et en faire un nouveau fichier "textefus.txt" sur l'unité C:

```
A>copy texte.txt+texte2.txt C:textefus.txt
1 File(s) copied
```

A>

Le fichier "textefus.txt" sera composé du fichier "texte.txt" suivi du contenu du fichier "texte2.txt".

```
A>type C:textefus.*
Ceci est un texte composé à partir
de la commande "copy con: texte.txt"
Cette commande peut donc être utilisée pour
créer des petits programmes ASCII sur disquette.
Seconde partie du texte
à fusionner avec "texte.txt"
A>
```

Dans le cas d'une fusion - qui s'adresse uniquement à des fichiers ASCII - le paramètre "/a" est pris par défaut. Si le fichier cible n'est pas spécifié, le premier fichier devient égal à son ancien contenu augmenté de celui du ou des fichiers donnés dans la commande.

EXEMPLE :

```
A>copy texte.txt+texte2.txt
      1 File(s) copied
```

Les deux fichiers seront concaténés sur l'unité A.

- On peut aussi fusionner un fichier avec une unité logique. Si nous voulons ajouter un texte au fichier "texte2.txt", par exemple, rien de plus simple :

```
A>copy texte2.txt+con:
Troisième partie entrée par con:
et ajoutée à "texte2.txt"
^Z
      1 File(s) copied
```

```
A>type texte2.txt
Seconde partie du texte
à fusionner avec "texte.txt"
Troisième partie entrée par con:
et ajoutée à "texte2.txt"
A>
```

----- En résumé -----

Changement d'unité implicite

Syntaxe	Exemples
<nouvelle unité>:	C: A:

COPY - Copie de fichier - Concaténation

Syntaxe	Exemples
copy <sour> <dest> copy < sour1>+<sour2>... +<sour _n > <dest> param. : /A /B /V	copy texte fichier copy *.bas C: copy C:prog copy p1/a+p2/b C:prog copy *.* C:/v

Commandes résidentes.