

# FLEXUTIL



Utilitaires d'exploitation



\*\*\*\*\*  
 \* UTILITAIRES FLEX \*  
 \*\*\*\*\*

Les utilitaires que vous trouverez sur votre disquette vous permettent des facilités d'exploitation et de développement appréciables.

Nous les avons classés en sept groupes, selon les services qu'ils vous offriront.

Le premier groupe comporte des utilitaires de gestion de fichiers, de catalogue et de supports magnétiques

PAGE		
6	---	CHECK . : compare deux fichiers
7	---	PDEL : détruit sélectivement les fichiers
8	---	ZAP : détruit tous les fichiers ou seulement certains
9	---	DUP : compare deux catalogues
10	---	REPLACE : effectue en même temps un DELETE et un RENAME
11	---	DIR : CATALOGUE détaillé
12	---	FILES : CATALOGUE simplifié (simple liste des fichiers)
13	---	FREE : indique la place disponible sur un support
14	---	INTEG : vérifie les secteurs libres d'une disquette

Le second groupe comporte des utilitaires qui accèdent à la mémoire centrale de l'ordinateur

16	---	MEMFILL : remplit la mémoire avec un caractère
17	---	CMEMEM : compare un fichier binaire et une zone mémoire
18	---	MEMOVE : recopie une zone mémoire dans la mémoire
19	---	DUMP : visualise le contenu d'un fichier, comme VISUMEM
20	---	OLOAD : charge un fichier binaire avec un déplacement
21	---	MEMEND : consulte la dernière position mémoire
22	---	RUN : déplace l'adresse d'exécution d'un programme

Le troisième groupe comporte des utilitaires de dépannage, d'installation et de test

24	---	RECOVER : restaure un fichier dans le catalogue
25	---	MAP : donne les adresses d'implantation et de lancement d'un fichier binaire
26	---	INSTALL : transforme un fichier .BIN en .CMD
27	---	FILETYP : donne la nature d'un fichier à partir de son contenu (Binaire ou ASCII)
28	---	TEST : vérifie les secteurs d'une disquette
30	---	MEMTEST : teste la mémoire centrale

Le quatrième groupe comporte des mots du langage de commande qui facilitent l'utilisation de l'utilitaire EXEC

RPT : répète une commande  
EXIT : envoie une chaîne de caractères sur l'écran  
HEXIO : envoie un caractère hexadécimal sur l'écran  
SLEEP : provoque un délai d'attente  
CONTIN : envoie un message 'CONTINUER ?' et gère la réponse

Le cinquième groupe comporte des utilitaires de travail sur des fichiers de type texte.

SPLIT : scinde un fichier texte en deux  
CONCAT : fusionne plusieurs fichiers texte en un  
EXTRACT : crée un fichier à partir de segments de fichiers

Le sixième groupe comporte des utilitaires de travail sur des fichiers générés par le Processeur de Texte.

REMSPC : enlève les blancs inutiles d'un fichier  
FIND : recherche les occurrences d'une chaîne  
WORDS : décompte les mots et les lignes d'un texte  
TYPOS : compte les occurrences de chaque mot d'un texte

Le septième groupe regroupe des utilitaires d'usage général

DATE : donne ou modifie la date du FLEX  
OKI : permet de fixer les paramètres d'une imprimante OKI  
(Version de PRINT.SYS)

## CHECK

La commande CHECK permet de comparer deux fichiers. Le résultat de la comparaison est affiché à l'écran.

### SYNTAXE :

CHECK,<nom du fichier 1>,<nom du fichier 2>

Le disque pris par défaut est le disque de travail.

L'extension prise par défaut est .TXT.

Les fichiers sont comparés octet par octet, leur contenu peut être quelconque.

### EXEMPLES :

réponse : CHECK,B.COPY.CMD,COPY.CMD  
Les fichiers sont identiques

réponse : CHECK,ESSAI,ESSAI1  
Les fichiers sont différents.

## PDEL

La commande PDEL est très pratique pour supprimer les fichiers devenus inutiles sur une disquette. L'utilitaire propose la destruction de tous les fichiers de la disquette désignée ou seulement ceux correspondant à un critère. La proposition de destruction est répétée pour chaque fichier.

### SYNTAXE :

PDEL|,<No de lecteur>||,<liste de chaînes cible>|

liste de chaînes cible : chaînes de caractères séparées par des virgules, correspondant aux premiers caractères des noms de fichiers concernés.

### EXEMPLES :

PDEL,1,ES.TXT,Z.BAS

réponse :      Effacement de "1.ESSAI.TXT"      (O/N) ? O  
                 Effacement de "1.ESSAJ.TXT"      (O/N) ? O  
  
                 Effacement de "1.Z.BAS"      (O/N) ? O  
                 Effacement de "1.ZDEFN.BAS"      (O/N) ? N  
                 Effacement de "1.ZDEMOL.BAS"      (O/N) ? O  
                 etc ...

Dans cet exemple, la commande propose la destruction de tous les fichiers du lecteur 1, qui commencent soit par ES, ayant l'extension .TXT, soit par Z et ayant une extension .BAS.

## ZAP

La commande ZAP est un utilitaire de destruction de fichier. L'on pourra à volonté, soit détruire tous les fichiers, soit spécifiquement détruire ceux indiqués par la liste de référence. La destruction d'un fichier se fait sans qu'il en soit donné avis. Cette commande est très utile pour supprimer d'un disque les fichiers devenus superflus.

### DESCRIPTION

La syntaxe en est :

```
ZAP|,<liste des lecteurs>||,<liste de référence>|
```

La "liste des lecteurs" et la "liste de référence" sont les mêmes que celles décrites à la commande "CAT". A l'exécution de la commande "ZAP", le nom de chaque fichier en cours de destruction apparaît à l'écran sous la forme :

EFFACEMENT DE "NO LECTEUR-NOM DE FICHIER"

Prennez garde à ce que vous ne pourrez pas revenir sur votre décision une fois l'ordre lancé. Lorsque l'utilitaire ZAP est appelé, les fichiers indiqués sont irrévocablement détruits sans intervention possible de l'utilisateur. Voici un exemple :

```
+++ZAP,1,.BAK
```

Cette commande détruit tous les fichiers du lecteur 1 comportant l'extension ".BAK". Aussi est il conseillé avant de lancer ZAP, de vérifier quels seront les fichiers détruits. Cette vérification se fera par l'intermédiaire des utilitaires CAT, DIR, ou FILES, sur la même liste de référence que celle qui sera appliquée à ZAP.

## DUP

La commande DUP permet de comparer le contenu des catalogues de deux disquettes. Cette commande affiche à l'écran la liste des fichiers qui se trouve dans un catalogue, et pas dans l'autre.

Cette commande est très utile pour comparer une disquette originale et une sauvegarde.

### SYNTAXE :

DUP,<No du lecteur 1>,<No du lecteur 2>

Cette commande établira la liste des fichiers qui sont sur le lecteur 1 (origine) et qui ne sont pas sur le lecteur 2 (cible).

### EXEMPLE :

DUP,1,Ø

réponse :	1.FLEXUTIL.TXT	Absent sur unité	No Ø
	1.ESSAI .BAS	Absent sur unité	No Ø
	1.Z .BAK	Absent sur unité	No Ø
	1.ZDEM01 .BAS	Absent sur unité	No Ø

## REPLACE

La commande REPLACE remplace un fichier par un autre. C'est-à-dire qu'elle supprime un fichier (ancienne version par exemple), puis qu'elle réattribue le nom ainsi libéré à un autre fichier (nouvelle version par exemple).

SYNTAXE :

REPLACE,<Nom de fichier 1>,<Nom de fichier 2>

où 'Nom de fichier 1' est le nom du fichier à supprimer et le nouveau nom que prendra le fichier intitulé 'Nom de fichier 2'.

Le lecteur par défaut est le disque de travail  
L'extension par défaut est .TXT.

Remarque : Cette commande est strictement équivalente à la séquence des deux commandes suivantes :

DELETE,<Nom de fichier 1>  
RENAME,<Nom de fichier 2>,<Nom de fichier 1>

EXEMPLE :

REPLACE,FICH1,FICH2

Réponse :

Le fichier "1.FICH1.TXT" doit-être effacer.!  
Vous confirmez (O/N) ? O

## DIR

L'utilitaire DIR est équivalent à la commande CAT mais il donne à l'écran tous les renseignements concernant chaque fichier.

Les renseignements fournis sont les suivants :

- No du fichier
- Nom du fichier
- Type d'extension
- Adresse de début d'implantation du fichier (Piste/secteur en hexa)
- Adresse de fin d'implantation du fichier (Piste/secteur en hexa)
- Nombre de secteurs occupés sur disque par le fichier
- Date de dernière mise à jour du fichier
- Type de protection affectée au fichier

EXEMPLE :

DIR,Ø,1,.SYS,F.TXT

On désire obtenir la liste des fichiers sur les drives Ø et 1 ayant l'extension .SYS et tous ceux commençant par la lettre F et ayant comme extension .TXT

Réponse :

Catalogue de l'unité Ø

Disque: G2FLEX |56 créé le: 5-Mar-82

NUMERO	NOM	TYPE	DEBUT	FIN	TAILLE	DATE	PRT
1	G2FLEX	.SYS	Ø1-Ø1	Ø1-19	25	2Ø-Nov-81	
2	PDIABLO	.SYS	Ø1-ØA	Ø1-ØA	1	14-Aut-81	
3	POKI	.SYS	Ø1-1B	Ø1-1B	1	18-Nov-81	
4	G2FLEX85	.SYS	Ø2-Ø7	Ø3-Ø1	25	3Ø-Dec-81	
15	F8PSFDD	.TXT	ØB-ØA	ØB-ØA	1	24-Dec-82	

Fichiers=139, secteurs=14Ø7, taille max.=1Ø8, libre 873

Catalogue de l'unité 1

Disque: DVP-FG |56 créé le: 5-Mar-82

NUMERO	NOM	TYPE	DEBUT	FIN	TAILLE	DATE	PRT
2	FLEXUT2	.TXT	Ø5-15	Ø6-Ø3	13	19-Mar-82	
9	FLEXUTIL	.TXT	Ø3-1B	Ø4-ØE	18	19-Mar-82	
11	FLEXUT1	.TXT	Ø4-17	Ø5-Ø2	1Ø	19-Mar-82	

Fichiers=8, secteurs=119, taille max.=28, libres=2161

## FILES

L'utilitaire FILES est équivalent aux commandes CAT ou DIR mais affiche uniquement le nom et l'extension de chacun des fichiers présent dans le catalogue.

SYNTAXE :

FILES|,|< No de lecteur >||,<liste de chaîne cible>|

liste de chaîne cible : chaîne de caractères séparés par des virgules correspondant aux premiers caractères des noms de fichiers concernés.

Les noms de fichiers seront listés en colonnes sur l'écran. Le nombre de fichiers par ligne dépend du paramètre de largeur de page du TTYSET. Si la largeur WD = 0, 80 colonnes sont supposées être disponibles et 5 noms de fichiers seront listés par ligne d'écran.

Une largeur d'écran inférieure entraînera un nombre moins important de fichiers listés par ligne d'édition.

EXEMPLES :

FILES,0,.SYS,F

Réponse :

Liste des fichiers de l'unité No 0

G2FLEX.SYS	G2FLEX85.SYS	ERRORS.SYS	POKI.SYS	PDIABLO.SYS
PRINT.SYS	G2FLEX18.SYS	ERRORSB.SYS	G2FLEXH8.SYS	G2FLEXHD.SYS
FILES.CMD	FIND.CMD	FREE.CMD	FILTYP.CMD	FLIST.CMD

## FREE

La commande FREE permet de connaître le nombre de secteurs libres sur une disquette. Le système affichera également le nombre de Kilo-octets restant disponibles.

SYNTAXE :

FREE|,<No de lecteur>|

Par défaut le numéro de lecteur pris est celui de travail.

EXEMPLE :

FREE,1

Réponse :

Secteurs libres : 2147  
Kilo-octets disponibles : 536

## INTEG

La commande INTEG permet de tester intégralement l'espace libre d'une disquette. (Secteurs non utilisés). Cet utilitaire garantit l'intégrité de l'espace disque non encore utilisé.

### SYNTAXE :

INTEG|,<No de drive>|

Le numéro de drive précise le disque qu'il faut tester.

L'extension par défaut est le disque de travail.

Cette commande vérifie que l'espace libre contient le bon nombre de secteurs, qu'il commence et termine aux bonnes adresses du disque. Si des différences ou des anomalies sont rencontrées, un message d'erreur approprié s'affichera.

Dans tous les autres cas le message suivant s'affichera :

L'espace disque est O.K !

### EXEMPLE :

INTEG,Ø

Cette ligne de commande va tester l'espace libre sur le disque du lecteur No Ø. Il faut noter que la commande INTEG nécessite un délai d'exécution assez long.

Toute disquette qui a passé le test INTEG avec des messages d'erreur ne devrait pas être utilisée pour créer de nouveaux fichiers.

Cette commande ne teste aucun espace disque qui soit occupé par des fichiers sur le disque.

## MEMFILL

La commande MEMFILL est utilisée pour remplir une ou plusieurs adresses mémoires avec une valeur particulière

Elle est utile pour certaines formes de développement et de mise au point de programmes.

### SYNTAXE :

MEMFILL,<Adresse de début>,<Adresse de fin>,<Valeur de l'octet>  
de remplissage

Les adresses doivent être précisées en hexadécimal. L'octet de remplissage est une valeur hexadécimale qui sera utilisée pour remplir les adresses mémoires comprises entre les bornes des adresses désignées. Si l'octet de remplissage est omis dans la ligne de commande, des zéros seront pris par défaut.

Après exécution, le contrôle sera rendu au FLEX, excepté s'il a été recouvert en écriture par la commande.

### EXEMPLE :

MEMFILL,C080,C09F,FF

Le tampon ligne du FLEX sera rempli entre les adresses \$C080 et \$C09F par la valeur hexadécimale \$FF.

## CMPMEM

La commande CMPMEM permet de comparer les contenus d'un fichier binaire sur le disque aux contenus de la mémoire où celui-ci devrait se charger.

Cette commande est utile pour la mise au point de programme ou pour détecter des problèmes mémoire éventuels.

SYNTAXE :

CMPMEM,<nom du fichier>

L'extension .BIN est prise par défaut.

Le fichier demandé sera lu exactement comme s'il devait être chargé en mémoire, sauf que chaque octet sera comparé à ce qui existe en mémoire. Si des différences apparaissent, elles seront imprimées ainsi que l'adresse, suivies des données en mémoire à cette adresse, puis des données du fichier disque.

EXEMPLE :

CMPMEM,VISUMEM.CMD

Le fichier VISUMEM.CMD du disque système sera lu et comparé au contenu des adresse mémoires normalement occupées par ce fichier.

## MEMOVE

La commande MEMOVE sert à transférer un bloc de mémoire centrale à n'importe quel autre emplacement de la mémoire.

### SYNTAXE :

MEMOVE,<adresse de début>,<adresse de fin>,<adresse de destination>

Toutes les adresses sont précisées en hexadécimal.

Les adresse de début et de fin désignent les bornes du bloc mémoire à déplacer, et l'adresse de destination désigne la position mémoire à partir de laquelle doit s'effectuer le transfert.

### EXEMPLE :

```
+++MEMOVE C080 C09F C0C0
```

Cette ligne de commande a pour effet de transférer une partie du tampon ligne du FLEX, de l'adresse C080 à l'adresse C09F, à l'adresse mémoire C0C0.

Le bloc de données original reste inchangé.

## DUMP

La commande DUMP est utilisée pour visualiser le contenu d'un fichier, secteur après secteur, en valeur hexadécimale et en code ASCII. Cet utilitaire permet de connaître le format exact d'un fichier sur disque par exemple, ou de visualiser son contenu.

SYNTAXE :

DUMP,<nom de fichier>

Par défaut l'extension prise est .BIN.

Chaque secteur affiché débute par deux valeurs hexadécimales qui représentent :

- Le premier le numéro de piste
- Le second le numéro du secteur en cours de visualisation.

Chaque ligne de données contient 16 chiffres hexadécimaux représentant les données, et sont suivies de leur représentation ASCII.

Tous les caractères ASCII inférieurs à \$20 ou supérieurs à \$7F sont représentés à l'affichage par un souligné.(-)

EXEMPLE :

DUMP Ø.VISUMEM

Le contenu de chacun des secteurs du fichier VISUMEM.BIN sera affiché sur l'unité périphérique choisie.

## OLOAD

La commande OLOAD est utilisée pour charger un fichier binaire en mémoire avec un décalage d'adresse d'implantation. Aucun code n'est modifié pendant le chargement. Cet utilitaire est pratique dans les applications de programmation en PROM.

### SYNTAXE :

OLOAD,<nom du fichier>,<valeur de décalage>

L'extension prise par défaut est .BIN.

Le disque pris par défaut est celui de travail

Le décalage disponible en option est une valeur hexadécimale qui devra être ajoutée à l'adresse d'implantation normale. Si le décalage n'est pas précisé, on choisit implicitement la valeur zéro ; ce qui mènera OLOAD à fonctionner exactement comme la commande GET.

L'adjonction du décalage bouclera l'adresse finale de \$FFFF. Par exemple, si un fichier est chargé normalement à l'emplacement \$60000 et qu'un décalage de \$A0000 a été précisé, le fichier sera nner exactement comme la commande GET.

L'adjonction du décalage bouclera l'adresse finale de \$FFFF. Par exemple, si un fichier est chargé normalement à l'emplacement \$60000 et qu'un décalage de \$A0000 a été précisé, le fichier sera chargé en \$00000.

### EXEMPLE :

OLOAD 0.VISUMEM 4000

Dans cet exemple, le fichier VISUMEM.BIN sera chargé en mémoire décalée à \$40000 de son adresse normale d'implantation. Au lieu de s'implanter en \$C1000, il s'implantera en \$01000.

Par conséquent, aucun code n'étant modifié, excepté si le fichier binaire est un code translatable, celui-ci ne s'exécutera pas à sa nouvelle adresse.

## MEMEND

La commande MEMEND sert à consulter la dernière position mémoire du FLEX, c'est à dire à relever sa valeur ou à la changer. Cet valeur est utilisée par de nombreux programmes FLEX afin de déterminer la dernière position qu'ils peuvent occuper.

### DESCRIPTION

La commande MEMEND a la syntaxe suivante :

```
MEMEND?  
ou  
MEMEND  
ou  
MEMEND <valeur>
```

"MEMEND?" affichera au terminal la valeur courante de la dernière position mémoire. "MEMEND" provoquera un test de la mémoire pour déterminer la plus haute adresse utilisable, test qui ne causera aucun dommage à la mémoire. Il est supposé que la mémoire de l'ordinateur est organisée en blocs mémoire de 4K commençant à la position 0. La valeur de la dernière position mémoire est celle de la dernière adresse contigue, déterminée par le test mémoire. Cette valeur est également affichée au terminal. "MEMEND <valeur>" attribuera à la dernière position mémoire la valeur spécifiée par le paramètre. Ledit paramètre devra être une valeur hex., et ne sera pas précédé d'un signe "\$". Voici quelques exemples :

```
+++MEMEND?  
+++MEMEND  
+++MEMEND BFFF
```

"MEMEND?" affichera au terminal la valeur courante de la dernière position mémoire. "MEMEND" provoquera un test mémoire qui déterminera la valeur de la dernière position mémoire contigue à la dernière adresse mémoire, et que cette dernière prendra. "MEMEND BFFF" modifiera la valeur de la dernière position de la mémoire, en lui faisant prendre la valeur "BFFF".

L'utilitaire MEMEND doit être utilisé chaque fois qu'un programme a modifié la dernière adresse de la mémoire, il fixera alors la valeur qu'elle doit prendre. C'est le cas d'une sortie anormale d'une procédure EXEC. EXEC modifie la dernière position de la mémoire pour se protéger, et s'il avorte, il ne pourra la restituer.

## RUN

L'utilitaire RUN sert à charger sur une adresse différente de celle où il s'exécute habituellement, un programme ne dépendant pas d'une position mémoire définie, et éventuellement l'exécuter.

### DESCRIPTION

La syntaxe en est :

RUN,<adresse de chargement>,<instruction>

ou

RUN/<adresse de chargement>,<instruction>

L'"adresse de chargement" désigne la position mémoire où l'instruction sera exécutée ; "instruction", l'instruction en question, avec ses paramètres. "RUN/<adresse de chargement>,<instruction>", chargera seulement l'instruction à l'adresse de chargement, sans l'exécuter. Les commandes reviennent alors au FLEX après chargement du programme. Veuillez noter que cet utilitaire ne modifie aucune instruction du programme. S'il s'agit d'un programme qui ne dépend effectivement pas d'une position mémoire, il s'exécutera correctement à la nouvelle adresse de chargement. Voici quelques exemples :

```
+++RUN,1000,FILES 0 .SYS
```

```
+++RUN/3500,TEST
```

Le premier exemple chargera le programme FILES à l'adresse 1000 hex. et en assurera l'exécution. Le deuxième exemple chargera le programme TEST.COM à l'adresse 3500 mais ne l'exécutera pas. En revanche, les commandes reviendront au FLEX.

## RECOVER

La commande RECOVER permet de restaurer un fichier sur disque dont le catalogue a été endommagé.

Il est nécessaire pour cette commande de connaître l'adresse de début d'implantation du fichier sur le disque.

### SYNTAXE :

RECOVER,<adresse disque>,<Nom du fichier>

L'extension prise par défaut est .TXT

Le nom du fichier donné est affecté au fichier une fois restauré sur l'unité de disque No Ø.

Le disque défectueux doit se trouver sur le lecteur 1.

Le lecteur Ø supporte le disque correct qui recevra le fichier après restauration.

L'adresse disque est donnée sous forme hexadécimale de deux nombres qui représentent le numéro de piste et le numéro de secteur à partir duquel s'implantait avant destruction le fichier à restaurer.

Cet utilitaire va lire sur l'unité 1 à l'adresse spécifiée les données qui s'y trouvent, puis les recopie sur l'unité de disque No Ø sous le nom attribué.

### EXEMPLE :

RECOVER Ø91A VISUMEM

Réponse :

Disque défectueux dans lecteur No 1 (O/N) ? 0

Cette ligne commencera à lire les données du lecteur No 1 à partir de la piste \$Ø9 et du secteur \$1A et les recopiera sur le drive Ø sous le nom de fichier : VISUMEM.TXT.

## MAP

L'utilitaire MAP permet de connaître les adresses d'implantation et de transfert d'un fichier binaire. Cette commande est utile dans un emploi associé à la commande SAVE.

### SYNTAXE :

MAP,<nom de fichier>

L'extension par défaut est .BIN.

Le disque affecté par défaut est le disque de travail.

Les adresses de début et de fin de chaque bloc de séquence résultante seront affichées sur le terminal.

Si une adresse de transfert est contenue dans le fichier, elle sera affichée à la fin de la liste d'adresses.

Si le fichier contient plusieurs adresses de transfert, seules l'adresse réelle (la dernière rencontrée) sera affichée.

### EXEMPLE :

```
+++MAP VISUMEM
```

### REPONSE :

```
C100-C1A1  
C100-C3E8  
C100
```

## INSTALL

L'utilitaire INSTALL est un moyen pratique pour renommer un fichier d'extension .BIN en .CMD. Le fichier préserve son nom initial. Par ce biais, vous 'installez' votre fichier dans l'ensemble des commandes utilitaires FLEX.

### SYNTAXE :

INSTALL,<nom du fichier>

L'extension du fichier par défaut est .BIN.  
L'unité de disque prise par défaut est celle de travail.

Cet utilitaire a la même application que la commande :

+++RENAME,FICHER.BIN,FICHER.CMD

ou que la commande :

+++ASMB,FICHER.TXT,FICHER.CMD

## FILTYP

La commande FILTYP sert à déterminer le type d'un fichier : BINAIRE ou TEXTE. Elle est utile lorsque des extensions de fichiers personnalisées ont été choisies et que l'on a oublié le type du fichier.

SYNTAXE :

FILTYP,<nom du fichier>

Par défaut, c'est l'unité de travail qui est sélectionnée

EXEMPLE :

+++FILTYP Ø.VISUMEM

REPONSE :

Fichier de type binaire

## TEST

La commande TEST est utilisée pour tester tous les secteurs d'une disquette. Tout secteur défectueux sera affiché sur le terminal.

SYNTAXE :

TEST|,<No du l'unité de disque>|

Par défaut, c'est l'unité de travail qui sera sélectionnée.

Tous les secteurs défectueux trouvés au cours du test sont reportés sur le terminal sous la forme de deux chiffres hexadécimaux :

- Le premier représente le numéro de piste
- Le second le numéro de secteur

La commande va tout d'abord lire le nombre de pistes et de secteurs maximum contenus sur le disque.

Si ces valeurs ne correspondent pas aux paramètres contenus dans la commande (cas des disques durs), le système affiche :

Piste/secteur maximum lu : 4C/FF  
Est-ce correct ? (O/N) :

> Si la réponse est 'O', le système effectuera la commande de test de la disquette assignée;

> Si la réponse est 'N', le système affichera les messages suivants :

Taille disque (5/8): 8  
35 ou 40 pistes (3/4):.....(uniquement dans le cas des  
disquettes 5 pouces.)  
Simple ou double densité (S/D): S  
Simple ou double face (S/D): S

De la même manière, si les informations secteurs sont invalides, le système affiche le message d'erreur suivant :

Secteur d'information système invalide

La commande TEST posera alors les mêmes questions que ci-dessus afin de pouvoir s'effectuer malgré cette défaillance système.

MESSAGES STANDARD :

- Secteur non trouvé ..... Secteur physiquement absent sur la disquette en test
- Erreur de CRC ..... Erreur dans le contrôle dans la parité du secteur en cours
- Chainage incorrect ..... Chainage des secteurs FLEX défectueux

## MEMTEST

La commande MEMTEST effectue un test de configuration de la mémoire centrale. Ce test détectera 99 % de tous les problèmes de mémoire si on lui octroie un certain temps d'exécution.

### SYNTAXE :

```
MEMTEST,<adresse de début>,<adresse de fin>  
de/test de test
```

L'adresse de début de test est une valeur hexadécimale qui précise l'adresse mémoire à partir de laquelle doit débiter le test.

L'adresse de fin de test est un nombre en hexadécimal qui détermine l'adresse mémoire ultime à tester.

Le test mémoire remplit l'espace ainsi déterminé par des valeurs aléatoires qui sont ensuite relues et comparées à celles qui avaient été précédemment écrites. Le processus se répète ensuite.

Pour que ce test soit efficace, il lui faut disposer d'au moins 4 heures d'exécution.

Il faut veiller à ne pas désigner une plage de mémoire recouvrant le programme de test lui-même.

Chaque passage réussi à travers le test sera marqué par un point d'exclamation ( ! ) qui s'affichera à l'écran.

Dans le cas où les valeurs lues ne sont pas identiques à celles précédemment écrites, le système affiche :

- L'adresse mémoire
- La valeur binaire de l'octet à l'écriture
- La valeur binaire de l'octet au moment de la lecture

### EXEMPLE :

```
+++MEMTEST,0,BFFF
```

Test mémoire de l'espace RAM utilisateur

### REPONSE :

```
ERR : 9BFF      ECR : 00010010      LEC : 00000010
```

Pour sortir de la procédure de test mémoire, il faut faire un RESET et recharger le système d'exploitation FLEX.

## RPT

La commande RPT permet de relancer l'exécution d'une commande autant de fois qu'il sera demandé.

Cette commande est utile dans les diagnostics et les démonstrations.

### SYNTAXE :

RPT,<nombre de répétitions>,<commande à exécuter>

Le nombre de répétitions donné en décimal spécifie le nombre de fois que la commande doit être exécutée.

La ligne de commande peut contenir n'importe quel utilitaire FLEX sauf RPT lui-même.

### EXEMPLE :

```
+++RPT 5 DIR Ø 1 .CMD
```

Cette ligne de commande a pour effet d'afficher 5 fois de suite les catalogues des unités de disque Ø et 1 dont les fichiers ont l'extension .CMD

## ECHO

La commande ECHO est utile pour l'incorporation dans des fichiers de commandes EXEC.

Elle permet de rendre l'écho de chaînes de caractères ASCII sur le terminal.

SYNTAXE :

ECHO,<chaîne de caractères>

EXEMPLE :

+++ECHO,LE PROCESSUS DE COPIE COMMENCE

REPONSE :

LE PROCESSUS DE COPIE COMMENCE

Il est souvent utile d'employer la commande ECHO dans les longs fichiers de commande EXEC pour envoyer des messages instructifs au terminal, afin d'informer l'utilisateur que l'opération est en cours.

## HECHO

La commande HECHO est utilisée pour envoyer des codes de contrôles en hexadécimal sur le terminal de son choix.

### SYNTAXE :

HECHO, <valeurs hexadécimales à envoyer>

### EXEMPLE :

+++P HECHO 0C,0D,0A,07

Dans cet exemple, la commande va envoyer à l'imprimante les codes de contrôles permettant d'effectuer :

- un saut de page
- un retour chariot
- un interligne
- puis un BIP sonore qui annoncera que la commande est terminée

Cette commande peut être utile dans le cas où l'on doit envoyer des caractères spéciaux à une imprimante pour changer le mode d'écriture, le nombre de caractères par pouce, le nombre de lignes par page. (Caractères compressés, étendus, 8 caractères par pouces ...).

## SLEEP

La commande SLEEP permet d'effectuer une temporisation lors de l'exécution de plusieurs utilitaires FLEX intégrés dans une commande EXEC.

### SYNTAXE :

SLEEP,N'

N : représente le nombre de secondes pendant lequel on veut "endormir" le système.

Il faut remarquer que pour une valeur importante, supérieure à 15, le délai peut ne pas être exactement respecté.

Une fois la commande SLEEP en cours d'exécution, il n'est pas possible de l'interrompre. Il faut attendre jusqu'à l'écoulement total du délai.

### EXEMPLE :

+++SLEEP,10

Cette commande aura pour effet de marquer une pause pendant environ 10 secondes.

## CONTIN

La commande CONTIN s'utilise pour les fichiers de commandes EXEC répétitifs ou complexes.

Elle demande une réponse par 'OUI' ou 'NON' suivant que l'on désire poursuivre l'exécution du fichier de commandes ou que l'on souhaite stopper son déroulement.

### SYNTAXE :

```
CONTIN
```

### REPOSE :

```
Continue (O-N) ?
```

### EXEMPLE :

Soit le fichier de type .TXT, construit avec la commande BUILD.CMD, et contenant les commandes suivantes :

```
DIR Ø .SYS  
DIR 1 .SYS  
CONTIN  
DIR Ø .CMD  
DIR 1 .CMD
```

Son exécution s'effectuera par la commande EXEC.

Le système donnera la liste des fichiers de l'unité Ø puis 1 dont l'extension est .SYS.

La commande CONTIN s'exécutera pour demander si l'on doit ou non poursuivre les commandes contenues dans le fichier de commandes.

Si la réponse est " O " : le système continuera d'exécuter les commandes du fichier

Si la réponse est " N " : le système abandonnera l'exécution du fichier de commandes et redonnera la main au système d'exploitation FLEX

## SPLIT

La commande SPLIT est utilisée pour couper un fichier en deux nouveaux fichiers à un numéro de ligne donné.

Son emploi est utile lorsqu'un fichier devient trop grand et est difficile à exploiter ou pour introduire une séquence de texte souvent utilisée dans un autre fichier.

### SYNTAXE :

```
SPLIT,<nom du fichier>,<nom du fichier>,<nom du fichier>,<N>  
          d'entrée          de sortie 1      de sortie 2
```

Le fichier d'entrée désigne le fichier qui est à scinder.

Le fichier de sortie 1 est le nom à assigner au premier groupe de lignes du fichier d'entrée.

Le fichier de sortie 2 est le nom à assigner au reste du fichier à couper.

N : est le numéro de ligne à partir duquel doit commencer le fichier de sortie 2.

Par défaut, les fichiers pris ont une extension .TXT ; et l'unité de disque désignée est celle de travail.

### EXEMPLE :

```
+++SPLIT ESSAI ESSAI1 ESSAI2 20
```

Cette commande permet d'écrire les lignes 1 à 19 du fichier d'entrée appelé ESSAI.TXT sur un fichier ESSAI1.TXT et les lignes 20 jusqu'à la fin sur un fichier appelé ESSAI2.TXT.

Le fichier original ESSAI.TXT reste inchangé.

## EXTRACT

L'utilitaire EXTRACT a pour finalité de créer un fichier à partir de différents fichiers ou de segments de fichiers. Il n'est pas nécessaire de copier les segments pour effacer les fichiers et les concaténer. Un fichier de commande est utilisé pour indiquer à l'utilitaire quelles lignes des fichiers doivent être lues et concaténées pour former le nouveau fichier. Les données brutes d'un texte à traiter peuvent être également copiées à partir du fichier de commande.

### DESCRIPTION

La syntaxe est la suivante :

EXTRACT,<nouveau fichier>,<fichier de commande>

"nouveau fichier" est la spécification du fichier en cours de création, "fichier de commande" est le nom d'un fichier texte comportant les instructions propres à l'utilitaire. Le "nouveau fichier" ne doit pas nécessairement exister au préalable. L'extension par défaut pour les deux fichiers est TXT. EXTRACT lit le fichier de commande, procédant à l'exécution des instructions dans l'ordre où elles figurent au fichier, il crée ensuite le nouveau fichier selon les instructions qui s'y rapportent.

### INSTRUCTIONS

Une instruction est constituée d'une ligne au fichier de commande. Elle commence en colonne 1 par une parenthèse fermée ")". Une ligne ne comportant pas à son début une parenthèse fermée en colonne 1, est considérée comme du texte à traiter, et est immédiatement copiée sur le fichier en cours de création. La structure d'une instruction est la suivante :

)<spécification du fichier><liste optionnelle des lignes>

La "spécification du fichier" cste à traiter, et est immédiatement copiée sur le fichier en cours de création. La structure d'une instruction est la suivante :

)<spécification du fichier><liste optionnelle des lignes>

La "spécification du fichier" concerne un fichier FLEX. TXT sera pris par défaut. Ce fichier doit bien sûr déjà exister. La "liste optionnelle des lignes" indique les lignes du fichier à lire et à copier. Il s'agit soit d'une série de numéros de lignes pris isolement, soit d'ensembles de numéros de lignes, ensembles séparés par une virgule.

Un ensemble de numéros de lignes est constitué de deux numéros séparés par un trait d'union, le premier représentant la première ligne, et le deuxième, la dernière. Si la première ligne n'est pas précisée, le début du fichier sera pris en compte. De même si la dernière ligne n'est pas donnée, la fin du fichier sera prise par défaut. Les numéros de lignes et les ensembles de numéros de lignes ne doivent pas forcément figurer par ordre alphabétique. Si nécessaire le fichier sera lu dans les deux sens afin d'atteindre les lignes à copier. Lorsque le dernier caractère d'une ligne de commande est une virgule, l'instruction est censée se continuer en colonne 1 de la ligne suivante. De la sorte les instructions peuvent figurer sur plusieurs lignes. Par exemple :

```
EXTRACT FICHSORT.TXT FICHENT.TXT
```

Cette commande provoque la création du fichier "NEW.TXT", à partir des segments de fichiers indiqués au fichier de commande "INPUT.TXT".

Supposons que le fichier de commande pour l'exemple plus haut comporte :

```
)FICHIER1,5,7-10,2-3,50-  
TEXTE SUPPLEMENTAIRE A INSERER  
)FICHIER2  
)FICHIER1,-10,15,20,  
30,40-80
```

Le fichier NEW comportera alors, dans l'ordre, les lignes 5, les lignes comprises entre 7 et 10, les lignes 2 et 3, et les lignes comprises entre 50 et la fin du fichier FILEONE.TXT ; la ligne "TEXTE SUPPLEMENTAIRE A INSERER" ; la totalité des lignes du fichier FILETWO.TXT ; et les lignes suivantes du fichier FILEONE.TXT : du début jusqu'à la ligne 10, les lignes 15, 20, 30, et les lignes comprises entre les lignes 40 et 80.

## REMSPC

La commande REMSPC supprime tous les espaces en excédent d'un fichier. Cet utilitaire est surtout destiné à être utilisé pour des fichiers de type .TXT ayant qui seront édités avec le Processeur de Texte T.S.C.

### SYNTAXE :

REMSPC,<nom du fichier initial>,<nom du fichier de sortie>

L'extension des fichiers choisie par défaut est .TXT.

L'unité de disque choisi par défaut est le disque de travail.

<nom du fichier initial> est le fichier à traiter en entrée.

<nom du fichier de sortie> nom que le fichier portera une fois la commande exécutée. Le nom du fichier ne doit pas déjà exister sur le disque de destination.

REMSPC va convertir toutes les occurrences de deux ou plusieurs espaces en un seul espace, sauf si la ligne début par une astérisque ( \* ) dans la colonne 1. (ligne de commentaire).

Ces lignes de commentaire passeront au fichier de sortie sans être modifiées.

### EXEMPLE :

```
+++REMSPC,SOURCE1,SOURCE2
```

Le fichier appelé SOURCE1.TXT sera lu sur le disque de travail, tous les espaces en excédent seront supprimés et le fichier sera réécrit sur un fichier appelé SOURCE2.TXT sur le disque de travail.

## FIND

La commande FIND est utilisée pour retrouver dans un fichier texte toutes les lignes contenant une chaîne de caractères spécifiée.

### SYNTAXE :

FIND,<nom du fichier>,<chaîne de caractères>

Le nom du fichier en entrée prend une extension .TXT

L'unité de disque prise par défaut est le disque de travail.

La chaîne de caractères peut être n'importe quel caractère imprimable (Codes ASCII de 32 décimal à 128) et doit se terminer par un retour chariot ou par une fin de ligne.

Lorsque la commande s'exécute, toutes les lignes contenant la chaîne de caractères spécifiée apparaîtront sur l'écran précédées du numéro de ligne correspondant. A la fin le système affiche le nombre d'occurrences de la chaîne dans le fichier spécifié.

### EXEMPLE :

```
+++FIND,ESSAI.TXT,le plus souvent
```

### REPONSE :

10.00=on le trouve le plus souvent possible dans ce texte

Total des occurrences : 1

## WORDS

La commande WORD sert à obtenir le nombre total de mots et de lignes d'un fichier de type texte. Il s'avère très utile pour la préparation d'un document ou d'un rapport, car il permet de garder une trace de la dimension d'un fichier.

### SYNTAXE :

WORDS,<nom du fichier>

Le nom du fichier prend par défaut l'extension .TXT

L'unité de disque sélectionnée par défaut est le disque de travail.

Sur exécution de la commande, WORDS va lire le fichier concerné, compter les mots et les lignes puis reportera les totaux sur l'écran du terminal. Un mot est défini comme un ensemble de caractères séparé d'un autre par un espace blanc ou un retour chariot.

### EXEMPLE :

```
+++WORDS,ESSAI
```

### REPONSE :

```
Nombre total de mot.. : 257  
Nombre total de lignes : 77
```

## TYPOS

L'utilitaire TYPOS sert à regrouper et à compter tous les mots d'un fichier. Il constitue une aide appréciable pour détecter les mots mal orthographiés ou les erreurs de typographie, ainsi que pour relever les répétitions trop fréquentes de mots dans un document.

### SYNTAXE :

TYPOS,<nom du fichier>|,<compter>|

<compter> spécifie la fréquence d'occurrence la plus élevée d'un mot qui devra être listée dans le décompte final des mots.

Le nom du fichier prend l'extension .TXT par défaut.

L'unité de disque choisie par défaut est celle de travail.

Si le compte n'est pas spécifié, l'affectation par défaut sera de 3, de sorte que les mots se répétant trois fois ou moins seront listés.

Toutes les lettres sont prises en minuscules, par conséquent les mots tels que "Test" et "test" seront considérés comme identiques.

Dans la liste finale des mots, chaque mot imprimé en minuscules sera précédé de sa fréquence d'occurrence.

Les mots les plus souvent employés seront imprimés en première position.

Les mots sont classés en ordre alphabétique à l'intérieur de chacune des fréquences.

### EXEMPLE :

+++TYPOS,ESSAI 20

Dans cet exemple seront dénombrés et affichés, tous les mots du fichier ESSAI.TXT apparaissant 20 fois ou moins.

## DATE

L'utilitaire DATE permet la mise en fonction et l'affichage du registre "date" du système. Le registre "date" est utilisé par d'autres utilitaires du FLEX. (Ex : NEWDISK...)

SYNTAXE :

DATE|,<spécification de la date>|

<spécification de la date> désigne les paramètres jour,mois,année. La date doit être rentrée sous la forme abrégée suivante : JJ.MM.AA. Si l'on omet ces paramètres dans la ligne de commande, le système affiche la date courante du système.

EXEMPLES :

+++DATE

+++DATE 21.04.82

Dans le premier cas le système affichera la date courante du système.

Dans le second cas, la date système sera modifiée par rapport à son contenu initial et deviendra le 21 AVRIL 1982.

## OKI

La commande OKI permet de fixer les paramètres internes d'une imprimante du type OKI 82 ou OKI 83. (Modes Compressé, normal...). De plus cette commande réinitialise à zéro le contenu du buffer de l'imprimante et initialise le compteur de ligne à zéro. (Equivalut au 'TOF SET')

### SYNTAXE :

OKI

### REPONSE :

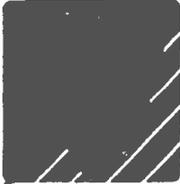
MODES : \*Compressé \*Normal \*Semi-etendu \*Etendu  
ESPACEMENT : \*6 lignes-pouces \* 8 lignes-pouces  
ESPAC. CAR.: \*<A> ligne courte \*<B> ligne longue

Les commandes disponibles correspondent au caractères majuscules qui suivent chacune des astérisques. Toute autre fonction est invalide. Pour sortir de la commande il suffit de frapper un retour chariot.

# FLEXUTIL



Utilitaires d'exploitation



**goupil**