

# PROLOGUE



**R 2 E**

zone d'activités de courtaboeuf

avenue de scandinavie

bp 73 - 91403 orsay

tél. : (1) 907 47 77

telex : r2e692014F



# PROLOGUE

EDITION DECEMBRE 79/JOUBERT/M-B



SOMMAIRE
----------

INTERFACE UTILISATEUR	3
INTERFACE OPERATEUR	36
COMMANDES DES LOGICIELS DE BASE	40
EDITEUR DE TEXTE	51
EDITION DE LIENS	57
MOMIC/Z	64
NOYAU DE GESTION DE FICHIERS	76
PARAMETRAGE DU SYSTEME	88

ANNEXES

- I - INSTRUCTIONS CLASSEES PAR MNEMONIQUES
- II - MEMENTO DES COMMANDES DU LOGICIEL DE BASE
- III - LISTE DES CODES ERREUR

## INTRODUCTION

PROLOGUE est un système de développement de programmes.

Il offre les services suivants :

- création et modification d'un fichier source
- assemblage
- résolution des liens entre différents fichiers objets, de manière à obtenir un fichier objet unique
- chargement et exécution d'un fichier objet
- possibilité d'intervention de l'opérateur lors de l'exécution grâce au moniteur MOMIC
- duplication de supports ou de fichiers

PROLOGUE met à la disposition du programmeur différents services tels que :

- utilisation du noyau de gestion de fichiers
- accès à la console et à l'imprimante
- analyse d'un nom de fichier
- chargement d'un fichier objet en mémoire
- accès direct à tout périphérique supporté.

## INTERFACE UTILISATEUR

L'interpréteur est un programme toujours résident permettant de faire la liaison entre l'opérateur et le système.

Il charge en mémoire les programmes de l'utilisateur et gère lui-même certains périphériques tels que écran-clavier-imprimante.

Pour les autres, il transmettra la demande de l'utilisateur en faisant les correspondances entre le numéro de ressource et son adresse de lancement.

De plus, l'interpréteur met à la disposition de l'utilisateur un certain nombre de modules dont les spécifications sont décrites dans ce chapitre.

JP INTR

Retour à l'interpréteur de commandes en fin d'exécution d'un programme.

Avant de demander à l'opérateur une nouvelle commande, l'interpréteur ferme tous les fichiers restant ouverts.



**CALL ES**

entrée H-L adresse du descripteur

sortie Z=1 E/S OK

A=0

Registres B-C, D-E et H-L conservés

L'utilisateur doit fournir le descripteur suivant :

NUMERO DE RESSOURCE	0
NUMERO D'UNITE	1
FONCTION DEMANDEE	2
ADRESSE DEBUT DE ZONE	3 *
O	4 5
NOMBRE DE SECTEURS	6
ADRESSE SECTEUR	7
SUR	8
3 OCTETS	9

\* Notation inversée

Fonctions

40 H lecture

80 H écriture

82 H écriture en prémarquage

CALL ES

entrée H-L adresse du descripteur

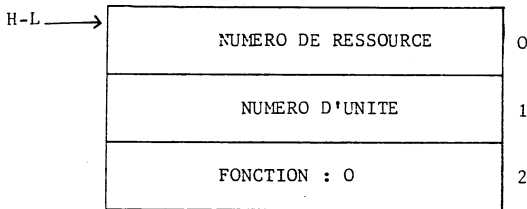
sortie Z=1 OK

A-B nombre de pistes sur le support

C nombre de secteurs de 256 octets par piste  
registres D-E et H-L conservés

Z=0 NOK A contient le numéro de l'erreur.  
registres D-E et H-L conservés

Le descripteur fourni par l'utilisateur est de la forme :



# ETAT D'UN DISQUE

CALL ES

entrée H-L adresse du descripteur

sortie Z=1 OK

A=0

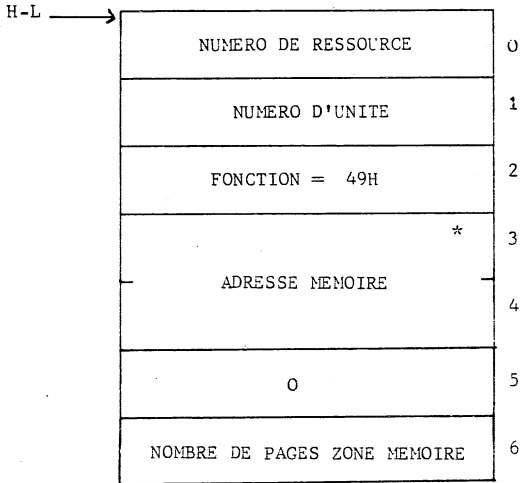
Registres B-C, D-E et H-L conservés

Z=0 NOK

A = numéro de l'erreur

Registres B-C, D-E et H-L conservés

Le descripteur fourni par l'utilisateur est de la forme.



\* Notation inversée

L'utilisateur reçoit dans sa zone mémoire les informations suivantes :

Zone Utilisateur →

NOMBRE TOTAL DE TRANSFERTS EFFECTUES SUR CETTE UNITE	0 1 2
NOMBRE D'ERREURS DE LECTURE	3 4
NOMBRE D'ERREURS D'ECRITURE	5 6
NOMBRE D'ERREURS DE POSITIONNEMENT	7 8
NOMBRE DE SECTEURS INVALIDES	9

		NUMERO DE LA DERNIERE ERREUR	10
		FONCTION	11
description de la dernière erreur survenue	-	NUMERO DE LA PISTE	12
			13
		NUMERO DU SECTEUR	14
adresse du premier secteur invalide	-	NUMERO DE LA PISTE	15
			16
		NUMERO DU SECTEUR	17
			18
//			
adresse du dernier secteur invalide	-	NUMERO DE LA PISTE	
		NUMERO DU SECTEUR	

Remarques :

- toute information est en notation directe (poids forts, poids faibles)
- les secteurs invalides décrits dans cette table ont tous reçu un secteur de remplacement. Cette opération n'a aucune influence sur l'utilisateur et la dimension du support reste inchangée pour l'utilisateur.

ANALYSE SYNTAXIQUE D'UN NOM DE FICHIER

---

CALL ANFC

entrée H-L adresse mémoire où se trouve la commande  
D-E adresse mémoire où l'on doit ranger le des-  
cripteur du fichier.

sortie Z=1 nom de fichier  
P = 1 avec support explicite  
P = 0 avec support implicite

CY=1 nom de support seul

dans ces deux cas → H-L pointe derrière le séparateur  
A contient le séparateur (VIRG ou CR)

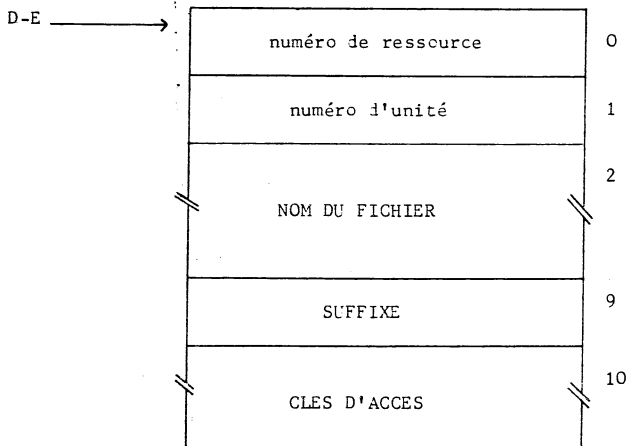
S=1 syntaxe incorrecte  
A contient le code erreur  
H-L conservés

Registres B-C et D-E conservés.

Un nom de fichier peut contenir dans l'ordre les éléments suivants :

	nom du volume	2 caractères
	numéro d'unité de ce volume	1 chiffre
séparateur → nom	.	
	nom du fichier	7 caractères maxi.
séparateur → suffixe	-	
	suffixe du fichier	1 caractère
séparateur → clés	:	
	cles d'accès en lecture et écriture.	4 caractères maxi.

Le descripteur formé par ANFC et dont D-E contient son adresse est donc constitué de la manière suivante :



Le descripteur formé par ANFC comporte donc 14 octets reportés de la manière suivante :

numéro de ressource	1 octet
numéro d'unité	1 octet
nom du fichier	7 octets
suffixe	1 octet
clés d'accès	4 octets

Toute information n'ayant pas été fournie par l'opérateur, est remplacée par des blancs ASCII dans ce descripteur.

Un nom de fichier ne peut pas comporter de caractères '.'.

Exemple de syntaxes correctes :

FLO.FICHER,	
FICHER,	fichier sur support implicite
FICHER-O,	type objet
FICHER-O : LCEC,	type objet + clés
FIC : LCEC,	clés
.FIC,	le '.' n'appartient pas au fichier

Exemple de syntaxes incorrectes :

FIC ;	séparateur incorrect
FICHERS,	trop long
F.IC ;	'.' dans le nom du fichier.



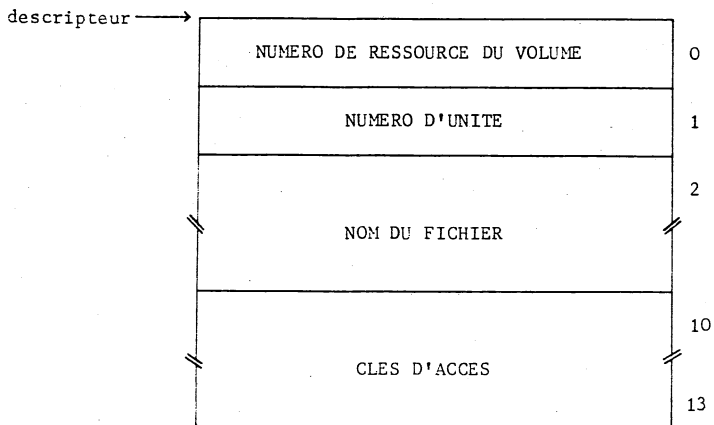
## CREATION ET OUVERTURE D'UN FICHIER

CALL CRFC

permet de créer un fichier, puis de l'ouvrir

CALL OPFC

permet d'ouvrir un fichier déjà existant



Entrée HL = adresse du descripteur

Sortie

registre B = numéro logique attribué à ce fichier

D-E = nombre de secteurs dans ce fichier

Registres C, H-L conservés

Erreurs pouvant survenir

fichier inexistant (OPFC)

fichier déjà existant (CRFC)

catalogue saturé (CRFC)

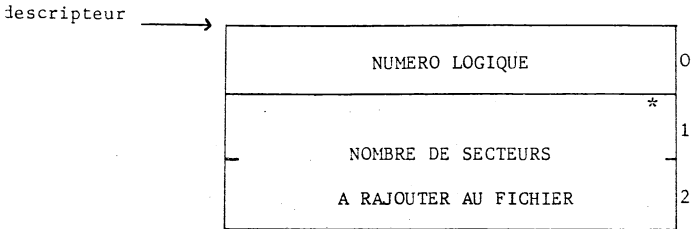
table des numéros logiques saturée

fichier déjà ouvert

AGRANDIR UN FICHER

CALL AGFC

agrandit un fichier de l'espace demandé



Entrée HL = adresse du descripteur

\* Notation inversée

Sortie Registres B-C, D-E et H-L conservés

Erreurs pouvant survenir

fichier fermé

fichier comportant trop de blocs d'extension

débordement du volume

Remarque :

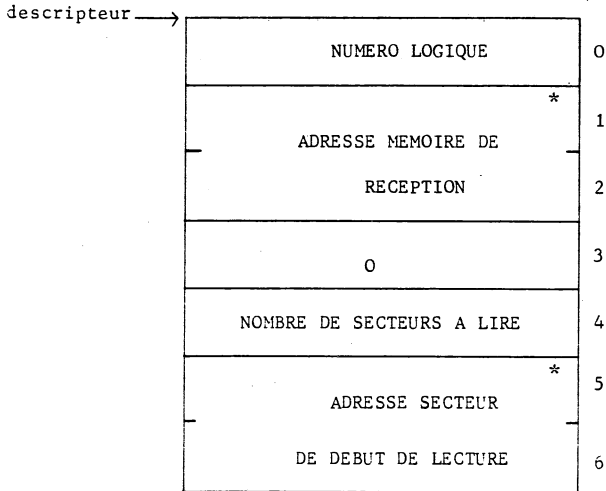
En dépit de l'allocation dynamique d'espace disque par le NGF, l'utilisateur peut souhaiter réserver à l'avance de l'espace disque au fichier :

- afin d'éviter toute impasse ultérieure quant à l'expansion de ce fichier sur le volume.
- afin d'éviter une trop grande dispersion de l'espace alloué à son fichier, dûe à l'expansion d'autres fichiers sur ce volume.

## LECTURE DANS UN FICHIER

CALL LCFC

permet de lire dans un fichier



\* Notation inversée

Entrée HL = adresse du descripteur  
Sortie Registres B-C, D-E et H-L conservés

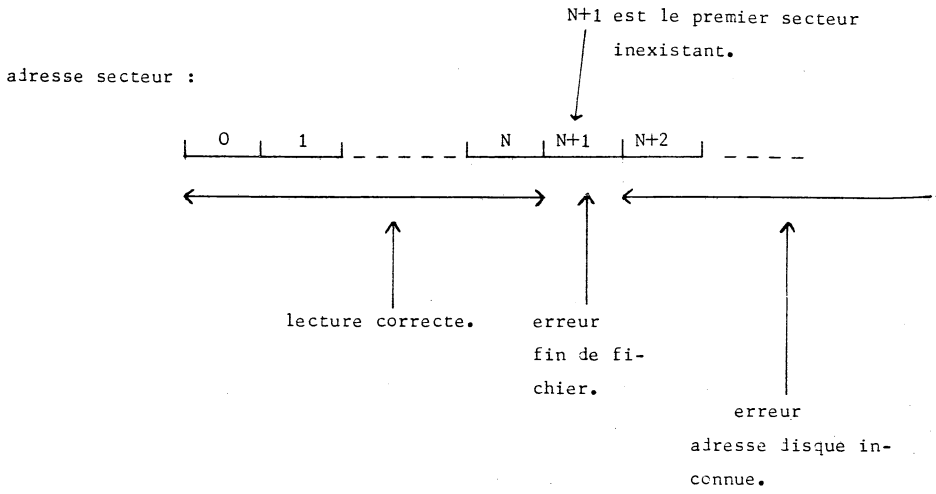
### Erreurs pouvant survenir

fichier fermé

fin de fichier : (tentative de lire le premier secteur qui se trouve en dehors du fichier)

adresse disque inconnue : (tentative de lire un secteur qui se trouve en dehors du fichier et qui est autre que le premier inexistant).

soit le schéma :

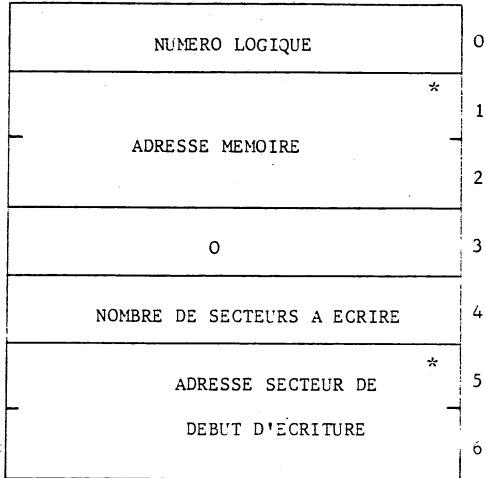


# ECRITURE DANS UN FICHIER

**CALL ECFC**

permet d'écrire dans un fichier

descripteur →



Entrée HL = adresse du descripteur

\* Notation inversée

Sortie Registres B-C, D-E et H-L conservés

## Erreurs pouvant survenir

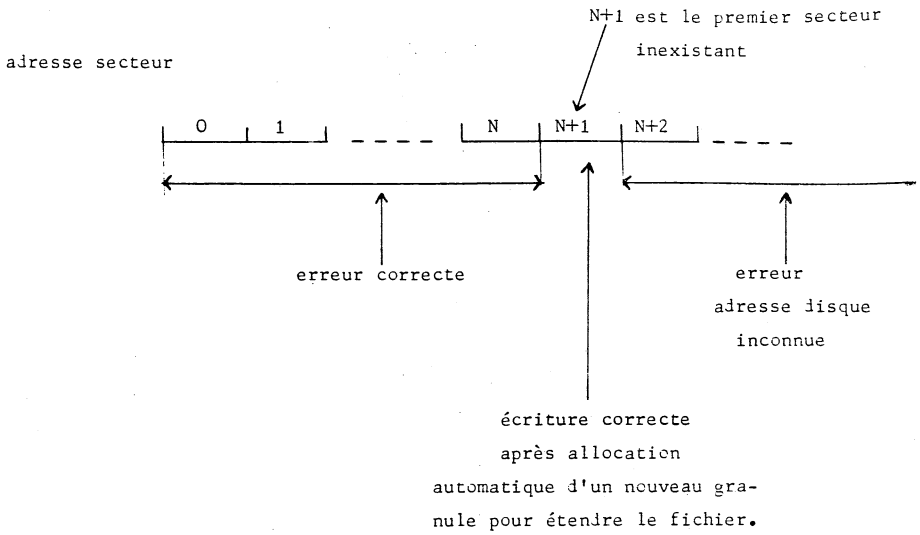
fichier fermé

fichier comportant trop de blocs d'extension

débordement du volume

adresse disque inconnue : (tentative d'écrire dans le deuxième secteur inexistant du fichier).

soit le schéma :



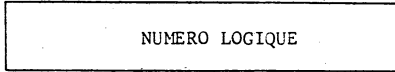
FERMER UN FICHIER

---

CALL CLFC

ferme un fichier

descripteur



0

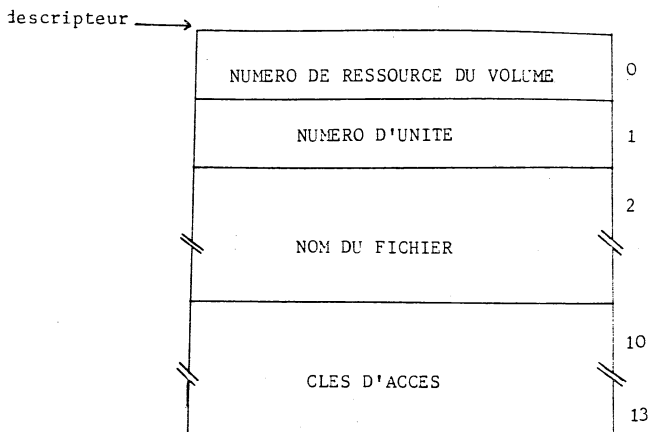
Entrée HL = adresse du descripteur

Sortie Registres B-C, D-E et H-L conservés

# DETRUIRE UN FICHIER

CALL SUFC

détruit le fichier demandé



Entrée HL = adresse du descripteur

Sortie Registres B-C, D-E et H-L conservés

Erreurs pouvant survenir

fichier inexistant

fichier déjà ouvert

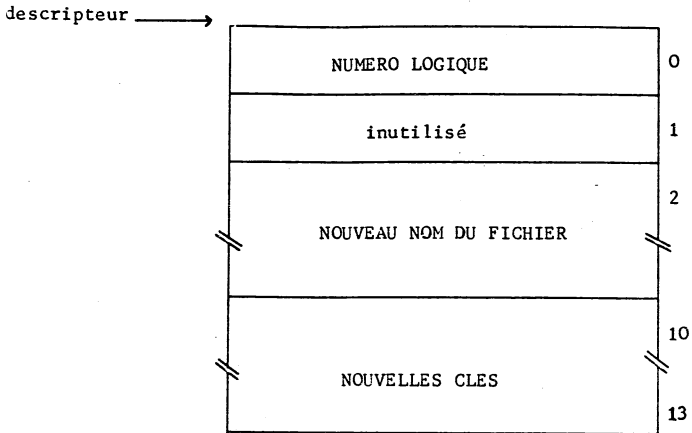
clés d'accès incorrectes



## RENOMMER UN FICHIER

**CALL RNFC**

permet de modifier le nom, le type ou les clés d'accès d'un fichier.



Entrée HL = adresse du descripteur

Sortie Registres B-C, D-E et H-L conservés

Erreurs pouvant survenir

fichier fermé

clés d'accès incorrectes

CHARGEMENT D'UN PROGRAMME EN MEMOIRE

---

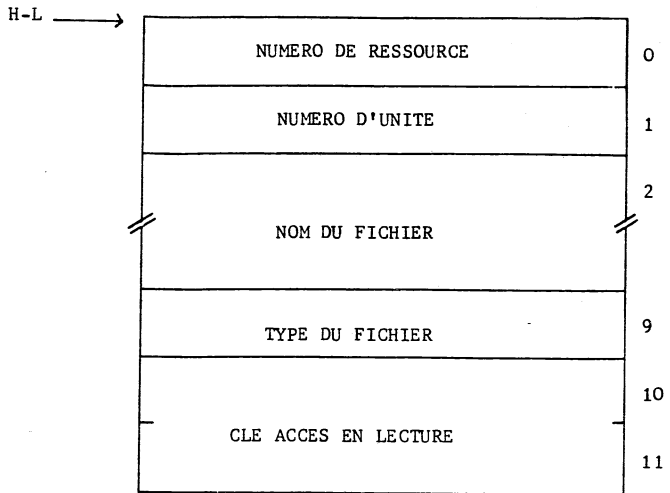
CALL CHGT

entrée	H-L	adresse du descripteur permettant de localiser ce fichier objet
	D-E	adresse d'implantation en mémoire
sortie	Z=1	fichier chargé DE= Adresse de lancement aucun registre conservé
	Z=0	A= Code de l'erreur survenue aucun registre conservé

Remarque :

CHGT charge tout fichier de type objet en mémoire.  
Il force le type 'O' (objet .

Le descripteur du fichier est le suivant :



Pour être chargeable, un fichier de type objet ne doit contenir aucun symbole externes non résolus.

DECODAGE ASCII → BINAIRE

---

CALL ABSN

entrée H-L      adresse de la valeur à décoder  
           C      valeur de la base utilisée

sortie D-E      valeur binaire  
           A et C    premier caractère hors base  
           H-L      incrémenté  
           Z = 1    valeur simple octet  
           CY=1    valeur double octet  
           S = 1    dépassement de capacité  
           B      conservé.

La valeur à décoder est une suite de caractères ASCII avec ou sans parité. La base utilisée ne peut être supérieur à 16 (valeur hexa-décimale).

Au retour, H-L pointe derrière le caractère séparateur :

Exemple :

	2 0 0 ,	A et C	contiennent le caractère ','
	↑            ↑	D-E =	valeur 200 si base décimale
H-L			512 si base hexadécimale
entrée	sortie		

CODAGE BINAIRE → DECIMAL ASCII

---

CALL BNDC

entrée D-E = valeur binaire  
H-L = adresse mémoire pour résultat

sortie H-L incrémenté  
registre B-C conservé

Remarques

- la valeur D-E est codée en ASCII 7 bits et rangée en mémoire
- au retour, H-L pointe derrière le dernier caractère
- les zéros non significatifs ne sont pas édités.

Exemple :

D-E contient la valeur 150

1 5 0  
↑    ↑  
H-L entrée    H-L sortie

- le nombre de caractères édités peut varier de 1 à 5 selon la valeur de D-E.

CALL BNHX

entrée     A =     valeur binaire  
           H-L =    adresse mémoire où l'on doit ranger le résultat

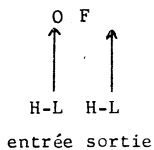
sortie     H-L     incrémenté  
           registres B-C et D-E conservés

La valeur A est codée ASCII 7 bits et rangée en mémoire à l'adresse H-L

Au retour, H-L pointe derrière le deuxième caractère :

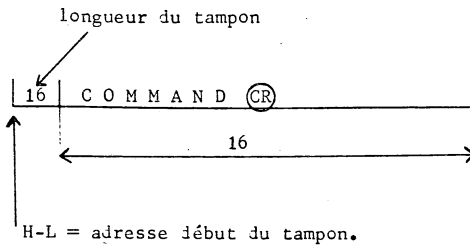
Exemple :

A contient la valeur décimale 15





Exemple :





CALL MES

entrée H-L = adresse mémoire où se trouve le message à éditer

sortie Registres B-C, D-E et H-L conservés

Remarque :

La fin du message est indiquée par le caractère ETX (03)

Le message à éditer doit être codé ASCII.

CALL EDER

entrée A = numéro de l'erreur

sortie Aucun registre conservé

édite sur l'écran le message suivant :

.ER. XX #

où XX est le numéro de l'erreur en représentation  
DECIMALE.

Une sonnerie est générée pour avertir l'opérateur.

TEST PRET CLAVIER

---

CALL TPCI

sortie Z = 0 clavier prêt, une touche a été enfoncée

Z = 1 clavier non prêt

Registres D-C, D-E et H-L conservés

ENTREE D'UN CARACTERE DU CLAVIER

---

CALL CI

sortie A = caractère reçu

Registres B-C, D-E et H-L préservés.

Remarque :

- si le bit numéro 7 (bit de signe) est positionné à 1, le caractère ESC précédait le caractère reçu.

Le caractère ESC est un paramètre de configuration du système.

- L'écho du caractère sur l'écran n'est pas effectué.

CALL CO

entrée C = caractère à éditer

sortie Registres B-C, D-E et H-L conservés

Remarque :

Si le bit numéro 7 (bit de signe) est positionné à 1, le caractère ESC est automatiquement édité avant l'édition du caractère contenu dans C.

Le caractère ESC est un paramètre de configuration du système.

CALL LO

entrée C = caractère à éditer  
sortie Registres B-C, D-E et H-L conservés

TABLE DES RELAIS DES FONCTIONS SYSTEMES

---

INTR	activation de l'interpréteur de commande retour à l'interpréteur de commande	1 1 0 H
ES	entrée/sortie directe sur un périphérique	1 1 3 H
ANFC	analyse syntaxique d'un nom de fichier	1 1 6 H
CRFC	création et ouverture d'un fichier	1 1 9 H
OPFC	ouverture d'un fichier	1 1 C H
AGFC	agrandissement d'un fichier	1 1 F H
LCFC	lecture d'un fichier	1 2 2 H
ECFC	écriture d'un fichier	1 2 5 H
CLFC	fermeture d'un fichier	1 2 8 H
SUFC	suppression d'un fichier	1 2 B H
RNFC	renommer un fichier	1 2 E H
CHGT	charger un programme	1 3 1 H
ASBN	décodage ASCII → BINAIRE	1 3 4 E
BNDC	colage binaire → décimal ASCII	1 3 7 H
BNHX	codage binaire → hexadécimal ASCII	1 3 A H
CLAV	acquisition d'une commande	1 3 D H
MES	édition d'un message sur écran	1 4 0 H
EDER	édition erreur sur écran	1 4 3 H
TPCI	test si prêt clavier	1 4 6 H
CI	entrée clavier d'un caractère	1 4 9 H
CO	sortie écran d'un caractère	1 4 C H
LO	sortie imprimante d'un caractère	1 4 F H

Remarque :

Pour utiliser ces points d'entrée, l'utilisateur doit déclarer en externe de son programme, les points d'entrée correspondants.

Les adresses indiquées à droite sont les adresses d'implantation des relais dans le cas où le système a été implanté à l'adresse 110H.

1 NOTATIONS UTILISEES

{ } l'une ou l'autre des réalisations au choix  
 [ ] caractéristique facultative, implicite par défaut

NOMSUP nom du support ainsi que le numéro de son unité.

NOMFIC nom du fichier. Il désigne aussi le support sur lequel se trouve ce fichier.

Il est structuré de la manière suivante :

NOMFIC = [NOMSUP.] nom du fichier [-type] [:cles]  
           ou     nom du fichier           1 à 7 caractères  
                   type                       1 caractère  
                   clés                        1 à 4 caractères

Le nom du support par défaut est le support utilisateur.

Le type et les clés, par défaut, sont des caractères blanc ASCII.

NOMPROG nom du programme

Désigne un fichier de type objet à charger et à exécuter. Il ne peut figurer qu'en tête de commande. Sa syntaxe est identique à celle de NOMFIC. Si le nom du support est omis, par défaut le support système est choisi.

v représente la touche validation de la commande. Il s'agit en général du retour-chariot.



## 2 SYNTAXE GENERALE D'UNE COMMANDE

L'interpréteur de commande prend en compte le chargement et le lancement de l'exécution d'un programme catalogué.

→ [& ADR,] NOMPROG, P1 ... , Pn v

Cette commande permet de charger en mémoire le programme NOMPROG tout en l'implantant à l'adresse ADR, puis lancer son exécution. Si l'adresse de chargement est omise, il est choisi l'adresse de chargement standard pour ce système (paramètre de configuration). Pour que ce chargement soit possible, NOMPROG doit être de type objet et ne doit pas comporter d'externes non résolus.

Lorsque l'exécution du programme est initialisée, le pointeur de pile (SP) adresse le premier paramètre de la liste, derrière la virgule qui fait suite au nom du programme.

S'il n'y a pas de paramètres, le pointeur de pile adresse le caractère 'RC' de fin de commande.

Pour récupérer le contenu du pointeur de pile, l'utilisateur peut utiliser la séquence suivante :

```
LD   HL, 0
ADD  HL, SP    HL = pointeur de la pile.
```

De plus, le stack fournit à l'utilisateur a une profondeur de 256 octets.

Remarque :

ADR est en notation hexadécimale, sans préfixe ni suffixe.

### 3 INTERFACE POUR LES LOGICIELS DE BASE

Les logiciels de base comprennent tous les outils facilitant le développement des logiciels d'application :

- édition du catalogue
- utilisation de conversions
- éditeur de textes
- assembleur
- édition de liens
- moniteur d'aide à la mise au point.

#### 3.1 Syntaxe des commandes des logiciels de base

L'appel d'un logiciel de base doit respecter la structure exigée par l'interpréteur des commandes.

Par ailleurs, les règles suivantes sont à observer :

- tout paramètre peut être précédé ou suivi de un ou plusieurs espaces, afin de donner plus de lisibilité à la commande.
- un paramètre numérique est exprimé en notation décimale, à moins qu'il ne soit précédé d'un caractère '/', auquel cas il est exprimé en notation hexadécimale.

Exemple : 15 expriment la valeur décimale 15.

/F

- il est possible de dissocier le chargement d'un logiciel de base et son exécution, ce qui permet de récupérer le support système pour y mettre un volume utilisateur.

Pour cela, il suffit de frapper le nom du programme sans donner les paramètres.

NONPROG v

Après chargement, une virgule apparaît à la place du caractère validation, permettant à l'opérateur de frapper la suite de la commande après avoir mis en place les volumes correspondant à l'application.

Exemple :

recopie d'une diskette (unité 0 à 1 sur floppy)

CP v DV, FLO, FL1 v



execution de la commande

chargement du programme de recopie à partir de la diskette système, puis mise en place de la diskette sur le support système avant exécution de la commande.

- Le rechargement du dernier programme exécuté peut être évité.

Il suffit qu'il commence par une instruction NOP (code 00) au point de lancement.

Dans ce cas le programme est simplement réinitialisé sans être rechargé.

Tous les logiciels de base, sauf MOMIC, respectent cette condition.

1 EDITION DU CATALOGUE D'UN SUPPORT

Il permet d'éditer sur l'écran ou l'imprimante la liste des fichiers contenus sur un support.

→ [NOMSUP1.] / [ , [NOMSUP2] [ ,LO ] ] v

si option LO présente : édition sur l'imprimante,  
les informations suivantes sont affichées :

- la longueur du granule en secteurs définie sur ce support.
- le taux d'occupation du support sous la forme NBMAX / NBOCC ou  
NBMAX = nombre total de granules  
NBOCC = nombre de ces granules occupés (affectés à un fichier).
- la liste des noms de fichiers avec leur suffixe, se trouvant sur ce support.

Exemples :

- /v                      affichage sur l'écran du catalogue du support utilisateur.
- /,FLO,LO v            impression sur l'imprimante du catalogue du floppy unité 0 .
- /,,LO v                impression sur l'imprimante du catalogue du support utilisateur.
- FL1./,DS2 v           édition du catalogue du disque 2. le programme est chargé à partir du floppy unité 1.

## 2 EDITION DU DESCRIPTEUR D'UN FICHIER

Il permet d'éditer sur l'écran ou sur l'imprimante le contenu du descripteur d'un fichier.

→ [NOMSUP1.] /, NOMFIC [LO] v

si option LO présente = édition sur l'imprimante.

Les informations suivantes sont affichées :

- le nombre de granules alloués à ce fichier
- le nombre de blocs que forme cette suite de granules.  
Un bloc est une suite de granules consécutifs.
- la description de ces blocs est fournie sous la forme :

A ---->(B - C) où

A est le numéro du bloc

B le numéro du premier granule constituant ce bloc

C le numéro du dernier.

### Exemples :

→ /, FICHIER-S v      fichier se trouvant sur le support utilisateur

→ /, FICHIER-S,LO v      impression sur l'imprimante

→ /, DSO.FIC-O v      fichier se trouvant sur le disque O.

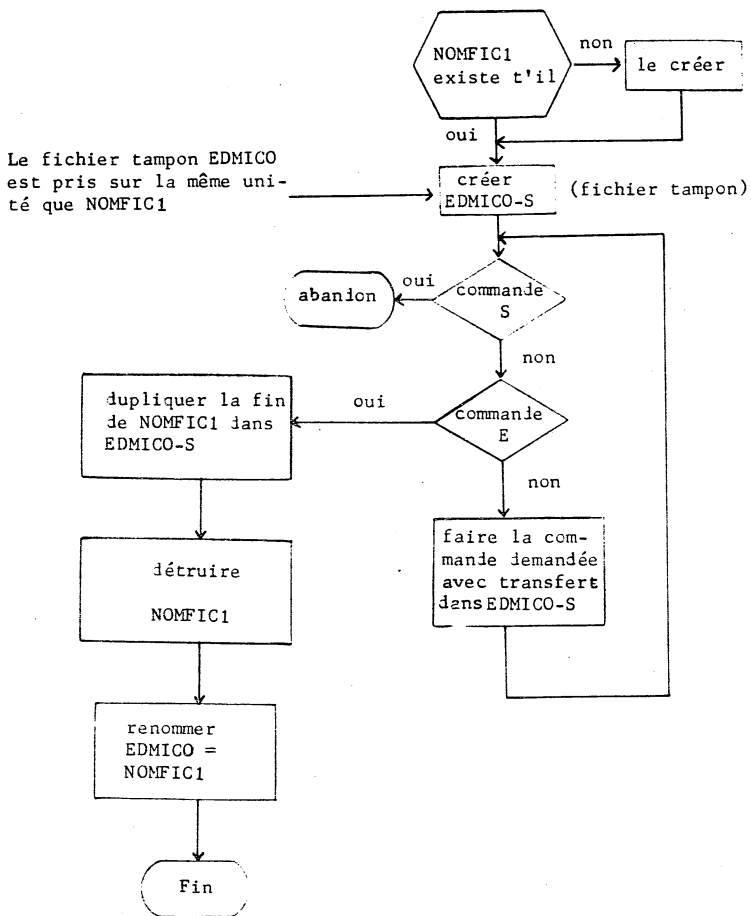
### 3 EDITEUR DE TEXTE

Il permet de modifier un fichier de type source ('S') ainsi que de le créer s'il n'existe pas déjà :

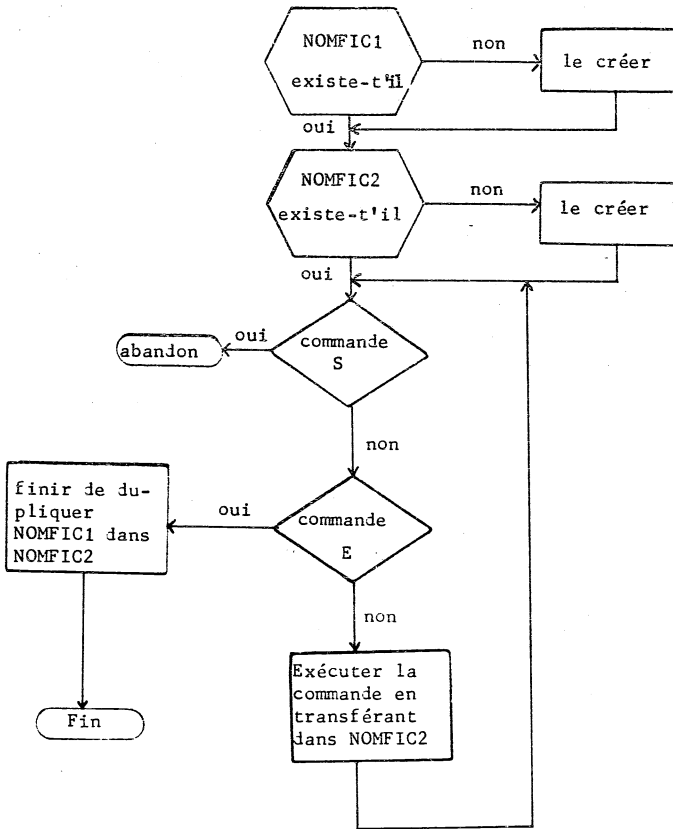
→ [NOMSUP1.] ED , NOMFIC1 [,NOMFIC2] v

Remarque Tous les fichiers doivent être de type source. Si le type est omis, NOMFIC1 et NOMFIC2 sont considérés de type source 'S'.

Organigramme de fonctionnement de l'éditeur, option NOMFIC2 ABSENTE



Organigramme de fonctionnement de l'éditeur, option NOMFIC2 PRESENTE



Exemples :

- ED , FIC v                    édition du fichier FIC se trouvant sur l'unité utilisateur.
- ED,FIC,DSO.NFIC v        le fichier NFIC est le résultat de l'édition. le fichier FIC reste inchangé.

#### 4 ASSEMBLEUR

Il assemble un fichier source et génère un fichier objet (voir notice ASMZ80.)

→ [NOMSUP1.] AZM,NOMFIC1 [ , { [NO  
NOMFIC2] } [,NL] [,NS] [,CO] ] v

avec NOMFIC1 le fichier source (type S)  
NOMFIC2 le fichier objet résultant (type O)  
NO pas de fichier objet demandé  
NL pas de listing  
NS pas d'édition de la table des symboles  
CO listing sur l'écran

- Si NOMFIC2 est absent, le nom du fichier objet sera NOMFIC1-O, sur le même support que NOMFIC1.
- Si CO est absent, l'impression se fera sur l'imprimante
- Le type des fichiers NOMFIC1 et NOMFIC2 est forcé respectivement à 'S' et 'O' par défaut.

#### Exemples :

- AZM,FIC v assemble le fichier source FIC et génère le fichier FIC-O sur le même support. Listing demandé.
- AZM,FIC,,CO v identique, mais génère le listing sur l'écran.
- AZM,FIC,NO,NL.v pas de fichier objet demandé .  
pas de listing demandé.



## 5 EDITION DE LIENS : EDL

EDL lie plusieurs fichiers objets entre eux et les réunit en un seul fichier objet résultant.

[NOMSUP.] EDL,NOMFIC1 [...,NOMFICn], = NOMFIC1 [,&ADR] [,LO] v

avec NOMFIC1 à NOMFICn : noms de fichiers objets à lier pour obtenir un seul fichier objet résultant. Les segments translatables sont placés les uns derrière les autres dans l'ordre de leurs déclarations.

Un nom de fichier peut être encadré de parenthèses auquel cas, seuls les points d'entrée de ce fichier sont pris en considération pour résoudre des symboles externes déclarés dans les autres fichiers.

Les fichiers doivent être de type 'O'

ADR : transforme le segment relogeable résultant en un segment absolu implanté à l'adresse ADR (notation hexalécimale).

LO : édition sur imprimante (sur écran en standard).

Se reporter, pour plus d'explications au chapitre 'Edition de liens'.

## 6 MONITEUR D'AIDE A LA MISE AU POINT

Il ne comporte pas de paramètres d'appel.

[NOMSUP.] MM v

MOMIC/Z est un logiciel d'aide à la mise au point.

Pour effectuer la mise au point d'un programme, il est possible de poser un piège à l'adresse /3F ; c'est le point de lancement de tout programme après chargement.

Si aucune adresse de chargement est modifiée, MOMIC viendra s'implanter à l'adresse implicite de chargement. Il peut être implanté ailleurs en mémoire par une commande du type :

→ & ADR,MM v

MOMIC/Z est chargé à l'adresse ADR.

Après chargement, MOMIC/Z devient indépendant du système.

L'ensemble des commandes est décrit au chapitre MOMIC/Z.

## 7 UTILITAIRE CP

### 7.1- Création d'un volume

[NOMVOL.] CP, CV, SUF, NOMVOL [ , [LGGR] , [NBFIC] , [ P ] ] v

initialisation d'un volume pour être géré par le système de fichier.

#### Paramètre de création

NOMVOL : nom du volume créé

LGGR : longueur du granule en secteur. Elle est une puissance de 2 et la valeur maximum est 256 secteurs. La valeur implicite est 16 secteurs.

NBFIC : nombre de fichiers présents dans le catalogue. La valeur maximum est 255 fichiers. La valeur implicite est 63 fichiers.

P : cette option demande le prémarquage du support. La demande de prémarquage est à confirmer par la réponse au message PREMARQUAGE (O/N) :

si l'option est absente, et si le volume à créer est déjà sous gestion de fichiers, le nom du volume existant est visualisé sous la forme NOMVOL (O/N) :  
où NOMVOL, est le nom du volume présent sur le support. En cas de réponse négative, le volume n'est pas altéré.

## 7.2 - Création de fichier

[NOMVOL.] CP, CF, IDFICH v

création d'un fichier sur un volume géré par le noyau de gestion de fichier. Aucun granule n'est affecté au fichier.

Le dépassement du nombre maximum de fichiers dans le volume est contrôlé.

## 7.3 - Suppression de fichier

[NOMVOL.] CP, SF, IDFICH v

suppression d'un fichier sur un volume géré par le noyau de gestion de fichier. Les granules affectés à ce fichier sont libérés et le nom disparaît du catalogue.

## 7.4 - Renommer un fichier

[NOMVOL.] CP, RF, IDFICH1, IDFICH2 v

permet de modifier les éléments de IDFICH1 par les éléments de IDFICH2.

Les éléments modifiables sont :

- le nom du fichier
- le suffixe du fichier
- les clés d'accès

Le nom du support doit être le même dans IDFICH1 et IDFICH2. Cette commande ne modifie pas le contenu du fichier renommé.

## 7.5 - Duplication de fichiers

[NOMVOL.] CP, DF, IDFICH1,  $\left. \begin{array}{l} \text{IDFICH2} \\ \text{SUF} \end{array} \right\} v$

a                      b

Recopie des données du fichier identifié par IDFICH1 dans le fichier identifié par IDFICH2.

Si le fichier identifié par IDFICH2 existe avant duplication, il est supprimé. Après recopie, les deux fichiers sont identiques.

Les identificateurs IDFICH1 et IDFICH2 doivent être différents.

Si le support est seul présent dans la partie b, le nom du fichier, le suffixe et les clés sont ceux donnés dans la partie a :

Exemple : CP, DF, FLO.FICH-S : 0102, FL1 v est équivalent à  
CP, DF, FLO.FICH-S : 0102, FL1. FICH-S : 0102 v

Remarque :

Les fichiers de sortie LO et CO sont respectivement l'imprimante et la console.

## 7.6 - Duplication de volume

[NOMVOL.] CP, DV, SUF1, SUF2 [I] v

duplication des fichiers du volume désigné par SUF1 dans le volume désigné par SUF2. Le volume SUF2 doit déjà être créé sous gestion de fichiers.

Chaque fichier présent dans SUF1 est recopié dans SUF2 par le même procédé que celui utilisé dans la duplication de fichiers (suppression préalable du fichier existant éventuellement dans SUF2). Au début de chaque duplication, le nom du fichier est visualisé à l'écran.

### Utilisation de l'option I

Tous les fichiers présents dans SUF2 sont supprimés avant de lancer l'opération de copie des fichiers.

Cette suppression est validée par une réponse O au message visualisé à l'écran

NOMVOL (O/N) : où NOMVOL est le nom du volume de réception.

Cette opération conserve les caractéristiques du volume :

- le nom du volume
- la longueur du granule
- le nombre maximum de fichiers dans le catalogue.

1 - GENERALITES

L'éditeur de textes permet de manipuler les fichiers sources :

- Ecriture d'un fichier source
- Modification, ajout ou suppression de lignes dans un fichier source.

Un fichier source est un fichier qui ne contient que du texte éditable. Ce texte est découpé en une suite d'enregistrements (lignes de texte) rangés sur le disque de façon séquentielle jusqu'à une marque fin de fichier. Tous les caractères du texte sont codés ASCII 7 bits. Un enregistrement se termine par un caractère 'Retour-chariot' et le caractère 'ligne suivante' soit la configuration ODH OAH.

Le dernier enregistrement du fichier se compose uniquement du caractère contrôle Z.

- Le symbole ':' à gauche de l'écran, indique que l'éditeur est en attente de commande
- La deuxième ligne visualisée est celle sur laquelle porte la commande
- Le passage à la ligne suivante provoque l'écriture de la dernière ligne
- Il est possible de corriger les erreurs de frappe dans une chaîne de caractères par les touches

← retour arrière 1 caractère  
 → progression 1 caractère  
 @ curseur en début instruction  
 ESC curseur en fin instruction.

- La première commande ne peut pas être de type M,C,Rou D tant que la première ligne n'a pas été lue.
- L'instruction ⊞ Z marque la fin du fichier.

## 2 - COMMANDES DE PROGRESSION DANS LE TEXTE :

: , chaîne v Recherche de la ligne qui contient la chaîne indiquée. Les blancs ne sont pas pris en compte (filtrés). La chaîne recherchée peut comporter 15 caractères au maximum.

Toutes les lignes lues sont écrites dans le fichier destination jusqu'à la ligne recherchée

Si la chaîne n'est pas trouvée, la recherche s'arrête en fin de fichier.



- :. chaîne v Identique à la commande précédente, la chaîne de caractères se trouvant en début d'instruction. Les blancs sont filtrés. Les lignes lues sont conservées.
  
- :D, chaîne v Recherche de la ligne qui contient la chaîne indiquée avec suppression des lignes lues. Cette commande et la suivante sont aussi appelées "commande d'extraction" car elles permettent d'isoler facilement un élément dans un texte.
  
- :D.chaîne v Identique à la commande précédente, la chaîne de caractères se trouvant en début d'instruction.
  
- :\* Recherche de la ligne contenant la chaîne précédemment définie. La recherche se fait avec conservation ou suppression des lignes lues, selon la nature de la commande précédente. Cette commande permet les recherches répétitives.
  
- :L Un espace permet de passer à la ligne suivante. La ligne précédemment visualisée est écrite.
  
- :Dn Suppression de n lignes de texte à partir de la ligne visualisée ( $0 \leq n \leq 9$ ). DO annule l'effet de la commande D.
  
- :A Ajout : positionnement en fin de fichier devant l'instruction Z en vue d'ajouter (par insertion) des éléments au texte original.

### 3 - COMMANDES D'INSERTION DE TEXTE :

:I Insertion d'un texte à la suite de la dernière ligne visualisée.

:..... instruction v

L'éditeur répond par 5 points en début de chaque ligne à insérer. L'utilisateur en fin de ligne valide son insertion par un retour-chariot ; ceci provoque le passage à l'insertion de la ligne suivante. La validation d'une instruction vide met fin à l'insertion de texte.

:J:NOMFIC Insertion d'un fichier derrière la dernière ligne visualisée.

La syntaxe de NOMFIC est celle d'un nom de fichiers.

L'insertion prend fin lorsque l'instruction Z est rencontrée. Cette dernière instruction n'est pas insérée.

Les fichiers source et destination doivent rester accessibles pendant la durée de l'insertion.

:R Insertion d'un texte avec suppression de la dernière ligne visualisée.

A part la suppression de la dernière ligne visualisée, cette commande a le même effet que la commande I.

#### 4 - COMMANDES DE MODIFICATION D'INSTRUCTION

- :M instruction y** Modification de l'instruction visualisée à partir du début de l'instruction.
- Le curseur se place sous le premier caractère de l'instruction. La progression dans l'instruction se fait par les 4 touches →, ←, @ et ESC. Tout caractère frappé vient en substitution du caractère correspondant dans l'instruction.
- Seuls les caractères à gauche du curseur seront validés après validation. En fin de modification, l'instruction reste présente sur l'écran et peut à nouveau être modifiée.
- 
- :C instruction y** Modification de l'instruction visualisée à partir de la fin de l'instruction.
- Cette commande équivaut à la commande M suivie de la frappe du caractère ESC. Le curseur est placé derrière le dernier caractère de l'instruction. Elle permet de compléter une instruction.
- Les touches ← et @ permettent de remonter dans l'instruction.

## 5 - COMMANDES FIN DE TRAVAIL

La commande Z met fin au travail.

Elle valide le travail effectué jusqu'au point où elle est introduite. Les lignes du fichier origine qui n'ont pas encore été lues sont annulées.

:E v      Fin de travail avec conservation des lignes non lues dans le fichier origine.

L'éditeur recopie toutes les lignes restant à lire du fichier origine vers le fichier destination jusqu'à la reconnaissance de la fin du fichier (caractère control Z).

:S v      Abandon du travail effectué. Le fichier origine n'a pas été modifié.

## 6 - COMMANDE L

:L : nn v

Permet de définir la longueur maximum d'un enregistrement ; place une butée avec alarme en cas de dépassement.

## 7 - TABULATIONS

Dans les commandes de modification ou d'insertion, la frappe de la touche tabulation permet un positionnement du curseur dans la ligne tous les 8 caractères.

1 - GENERALITES

L'éditeur de liens permet de regrouper des fichiers objets produits par l'assembleur pour former un nouveau fichier objet. Cette opération de regroupement s'accompagne :

- de la résolution des références entre fichiers (ENTRY/EXTRN)
- de la fixation éventuelle d'une adresse d'implantation
- de la visualisation de la structure du fichier résultant.

2 - DESCRIPTION DES CARTES COMPOSANT LE FICHIER OBJET

Le fichier objet comporte un certain nombre d'unités logiques : les cartes.

2.1 Carte P (programme)

c'est la première carte du fichier. Elle marque son début.

2.2. Carte E (Entry)

Chaque symbole défini comme point d'entrée (ENTRY) dans le programme est mémorisé dans une carte. Ces cartes permettent aux autres fichiers objets de résoudre leurs références externes. Les cartes E suivent la carte P.

2.3 Carte B (binaire)

Elle contient le code binaire du programme assemblé. Ce code peut être relogeable ou absolu suivant la description du programme source.

2.4 Carte R (relocation)

Elle permet d'indiquer la position des opérandes relogeables et externes dans le code binaire.

2.5 Carte X (externe)

Chaque symbole défini comme externe dans le programme est mémorisé sur une carte. Ces cartes indiquent les références qui sont à rechercher dans les cartes E des autres fichiers objets.

2.6 Carte F (fin)

Elle indique la fin du fichier objet.

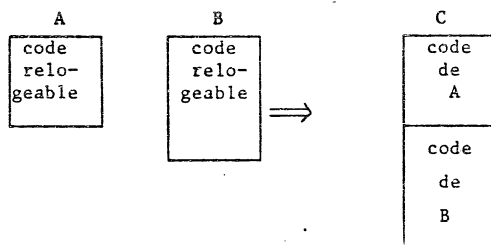
### 3 - ROLE DE L'EDITEUR DE LIENS

#### 3.1 Description générale

soit l'opération suivante : EDL A,B → C

signification : édition de liens entre les fichiers objets A et B, le résultat est rangé dans le fichier objet C.

Le code binaire relogeable de B est rangé après le code binaire de A pour donner un code binaire relogeable dans C. L'ordre des fichiers est respecté.



Les codes binaires absolus gardent leur implantation d'origine.

Les références externes de A et B sont résolues par les références points d'entrée (ENTRY) correspondantes. Les références externes non satisfaites sont recopiées dans le fichier C, ceci pour permettre leur résolution dans une autre édition de liens.

Remarque : un fichier comportant des références externes non satisfaites ne peut être chargé en mémoire pour être exécuté.

Les références points d'entrée de A et B sont recopiées dans le fichier C, ceci pour permettre à d'autres fichiers binaires de résoudre leurs références externes dans une autre édition de liens.

Remarque : si plusieurs points d'entrée de même nom apparaissent, seul le premier est retenu.

A la fin de l'édition de liens, le fichier C possède une structure de fichier objet, il peut donc participer à de nouvelles éditions de liens.

#### 3.2 Cas particulier

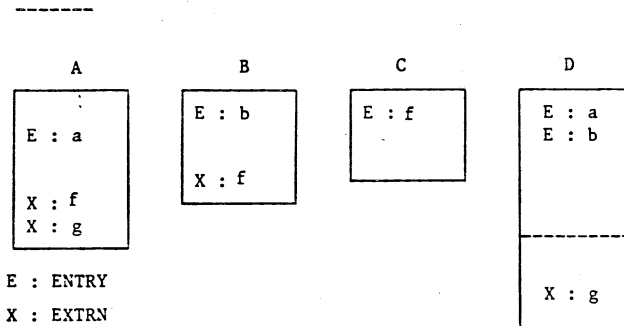
##### 3.2.1 Présence d'un fichier parenthésé

soit l'opération suivante : EDL A,B,(C) → D

Le code binaire du fichier C n'est pas rangé dans le fichier résultat D.

Seuls ses points d'entrée absolus sont utilisés pour résoudre les références externes des fichiers A et B. Cette fonction permet de résoudre facilement les liens avec un système d'exploitation.

Exemple :



La référence f est résolue grâce à la présence du fichier C. Elle n'apparaît plus dans la liste des EXTRN de D. Le code binaire de C n'est pas rangé dans D. Il est à remarquer que le point d'entrée F n'est pas mémorisé dans le fichier résultat.

### 3.2.2 Présence d'une adresse d'implantation.

soit l'opération suivante : EDL A,B→D adr = XXXX

Le fonctionnement est semblable au cas général à la différence près que le code binaire rangé dans le fichier D est absolu et implanté à partir de l'adresse indiquée dans l'opération.

## 4 - VISUALISATION DE LA STRUCTURE DU FICHIER RESULTAT ET DE SES PARAMETRES

Elle fournit :

- la liste des symboles points d'entrée provenant des différents fichiers regroupés à l'exception des fichiers parenthésés. Chaque symbole est suivi de son adresse d'implantation dans le fichier résultant. Le symbole est précédé d'un M s'il est défini comme ENTRY dans plusieurs fichiers,
- la liste des symboles externes restant à résoudre.
- l'implantation des différents segments du fichier objet résultat. Les bornes sont notées en hexadécimal. La présence d'un signe + devant chacune d'elle indique un segment relogeable. Le premier segment est accompagné du nom du fichier objet d'origine.

exemple :            [ + 0000, +0013 ]            OBJ-A  
-----            [   0200,   0203 ]

Le fichier OBJ-A a produit un segment relogeable et un segment absolu.

Si deux segments absolus se chevauchent, l'indication de conflit apparaît.

- L'adresse de lancement du programme résultat.

Elle est signalée par '---' et sa valeur peut être absolue ou relogeable. La valeur absolue 0000 signifie l'absence d'adresse de lancement.

## 5 - ERREURS DETECTEES PAR L'EDITION DE LIENS

### 5.1 Les erreurs fatales

Elles conduisent à l'arrêt de l'édition de liens. La liste est portée en annexe I.

### 5.2 L'erreur dans la résolution des références externes.

Si la valeur du point d'entrée ne peut être rangée dans le code binaire pour des raisons de volume, un message apparaît sous la forme :

aaaa { T }    nom    DEBORD. VALEUR  
      { A }

avec aaaa : adresse de rangement dans le code

      { T } : type de l'adresse (relogeable ou absolue)  
      { A }

nom : nom de la référence concernée.

La valeur 0 est rangée dans le code binaire.

### 5.3 Les erreurs apparaissant dans la visualisation finale.

En cas de multidéfinitions d'un symbole point d'entrée, seul le premier est mémorisé dans le fichier résultat.

En cas de conflit d'implantation entre des segments absolus, le recouvrement se fait au moment du chargement pour l'exécution.



VISUALISATION DES ERREURS DANS LA RESOLUTION DES VALEURS DES EXTRN.

---

Dans le cas où le code binaire prévoit une valeur d'EXTRN sur un octet et qu'au moment de l'édition de liens celle-ci fait 2 octets, le message suivant est visualisé sur la console opérateur. La valeur 0 est rangée dans le code binaire

XXXX  $\left\{ \begin{array}{c} A \\ T \end{array} \right\}$  NOM SYMBOLE DEBORD. VALEUR

XXXX est l'adresse de rangement de la valeur de l'octet provenant de l'EXTRN (adresse dans le fichier résultant)

$\left\{ \begin{array}{c} A \\ T \end{array} \right\}$  type de l'adresse concernée (absolue ou relogeable)

NOM SYMBOLE donne le nom de l'EXTRN concerné.

DEBORD. VALEUR message indiquant l'erreur.

LISTE DES ERREURS DE L'EDITEUR DE LIENS

TABLE SATUREE	la mémoire utilisateur est inférieure à 2 pages.
TABLE ENTRY/EXTRN SATUREE	la table des symboles ENTRY et EXTRN at- teint la fin de la mémoire disponible par l'utilisateur.
TABLE MAP SATUREE	la table contenant les descriptions des segments de code qui composent le fichier résultant est saturée (30 segments sont prévus).
NB. FICH. EN ENTREE 9	le nombre de fichiers à grouper par l'édi- tion de lien est supérieur à 9.
TYPE DU FICH. INCORRECT	l'un des fichiers n'est pas de type "O" (objet).
VAL. ADR. D'IMPLANT. ERRON.	la valeur d'implantation donnée dans la commande n'est pas de la forme &XXXX où XXXX est une valeur hexadécimale.
PLUSIEURS ADR. DE LANCEMENT	Deux fichiers au moins réunis par l'édi- tion de liens contiennent une adresse de lancement.
ABS. FIN FICHER OBJET	l'un des fichiers objets cités ne possè- dent pas de marque de fin.

EXEMPLE DE VISUALISATION APRES EDITION DE LIENS

après la commande :

→ EDL, BPS1, BINTR, BFLD2, BNGF, = OSYST, &110, LO v

R2E EDL/Z80 - V1.0 = OSYST-0

LISTE DES PARAMETRES ENTRY

AGFC	OFC4	ANFC	O116	ASBN	O11C	BNDC	O11F	BNHX	O122
CHGT	O119	CI	O131	CLAV	O125	CLFC	OFAE	CO	O134

IMPLANTATION (S)

[0048 , 009E]	BPS1-0
[0110 , 013F]	BINTR-0
[003F , 003F]	
[0140 , 0627]	
[07E8 , ODAA]	BFLD2-0
[ODC5 , 17EE]	BNGF-0
→ , 0110	

## 1 - PRESENTATION

Le moniteur du Micral est un programme destiné à aider l'utilisateur dans la mise au point de ses applications.

Ses différentes commandes permettent :

- de lancer l'exécution d'un programme en demandant un ou plusieurs points d'arrêts.
- de suivre l'exécution d'un programme instruction par instruction
- de visualiser ou de modifier l'état des registres et des indicateurs du processeur
- de visualiser ou de modifier un ou plusieurs octets en mémoire

### Généralités sur MOMIC/Z

-----

Le moniteur, sous une présentation unique, est compatible processeur 8080 et Z 80.

Le processeur Z80 est un sur-ensemble du processeur 8080.

Il comporte 9 registres supplémentaires dont 2 registres d'index, et un jeu de 158 instructions au lieu de 78 sur le 8080.

Quel que soit le processeur utilisé, le moniteur offre les mêmes possibilités d'utilisation, chaque commande s'adaptant aux caractéristiques de l'un ou l'autre des processeurs.

## 2 - ADRESSAGE DIRECT OU RELATIF

MOMIC/Z dispose de 2 modes d'adressage de la mémoire :

- adressage direct
- adressage relatif

En adressage direct, l'adresse fournie par l'opérateur correspond à l'adresse réelle (physique) dans la mémoire.

En adressage relatif, l'adresse fournie par l'opérateur est ajoutée à l'adresse d'implantation du programme pour donner l'adresse réelle à atteindre.

L'adresse d'implantation du programme est donnée par la commande '& :'

L'adressage relatif est particulièrement adapté aux programmes relogeables.

Pour ceux-ci, l'utilisateur conserve l'adressage de son listing d'assemblage quelle que soit l'adresse réelle d'implantation.

Syntaxe :

-----

ADR : adressage direct

+ADR : adressage relatif

ADR est une adresse 16 bits en notation hexadécimale sans préfixe ni suffixe.

Toutes les commandes du moniteur admettent les 2 modes d'adressage.

Exemple :

Base &:1000

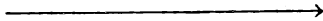
Adressage

Mémoire réelle

1001



+ 3



1000

1001

1002

1003

### 3 - LES COMMANDES DE MOMIC/Z

Format général des commandes :

Les éléments entre crochets sont facultatifs et dépendent de la commande. La fin d'une commande est marquée par le caractère 'rc' (retour chariot).

- COM : [FAC1] , .... [,FACn]

COM : caractère ASCII précisant le type de la commande

FAC1 - FACn : facteurs de la commande

La virgule est le seul séparateur autorisé entre les facteurs d'une commande.

Si une commande est syntaxiquement incorrecte ou si elle est invalide, le moniteur le signale par un caractère '#'.

La liste des commandes du moniteur figure en annexe au chapitre.

La commande A :

-A:

. Visualisation et modification du contenu du premier jeu de registres et d'indicateurs du processeur.

Les registres et les indicateurs présentés sont dans l'ordre :

A - B - C - D - E - H - L - CY - Z - S - P - R

soit 7 registres et 5 indicateurs ; l'indicateur addition/soustraction propre au Z80 n'est pas présenté.

La valeur des registres est représentée par 2 caractères hexadécimaux suivis d'un tiret.

La valeur des indicateurs est représentée par 1 caractère binaire (0 ou 1) suivi d'un tiret.

La présence du tiret donne à l'utilisateur, la possibilité de modifier la valeur imprimée, en frappant 1 ou plusieurs caractères hexadécimaux. (Seuls les 2 derniers caractères frappés sont pris en compte).

La frappe d'un espace permet de passer à l'impression de l'octet suivant.

La commande se termine après la visualisation de l'indicateur R (demi-report) ou après la frappe d'un rc.



La commande B :                    -B:  
-----

. Visualisation et modification du contenu du deuxième jeu de registres et d'indicateurs de processeur (Z80 uniquement)

Les registres et les indicateurs présentés sont dans l'ordre :

A' - B' - C' - D' - E' - H' - L' - CY' - Z' - S' - P' - R'.

L'utilisateur de la commande est identique à celle de la commande 'A:'

La commande S :                    -S:  
-----

. Visualisation et modification du contenu du registre pointeur de stack (8080 et Z80) et des registres d'index et vecteur interruption (Z80 seulement)

Les registres et les indicateurs présentés sont dans l'ordre :

SP - IX - IY - I

Les registres 16 bits (SP - IX et IY) sont présentés dans l'ordre normal poids forts, poids faibles.

La commande K :

-K:

-----

.Visualisation et modification du contenu de l'octet des clés (8080 et Z80) et de l'état de l'horloge temps réel ainsi que de l'état de masquage des niveaux d'interruption (Micral 8080 seulement).

Le premier octet précise le périphérique sélectionné à l'édition :

00 → visu  
01 → imprimante

Il autorise le Dump mémoire en mode page à page ou la liste de secteurs disque en mode secteur par secteur :

00 → continu  
80 → liste page à page ou secteur par secteur.

Les deux caractéristiques peuvent être combinées.

Le deuxième octet est l'octet d'état de l'horloge temps réel présent sur le Micral S (adresse 0F4H)

Le troisième octet est l'octet d'état de masquage particulier des niveaux d'interruption présent sur le Micral S (adresse 0F5H).

La commande D :           -D:FAC1,FAC2  
-----

. Liste de la zone mémoire commençant à l'adresse FAC1 et se terminant à l'adresse FAC2.

La liste (dump) apparaît à raison de 4 groupes de 4 octets par ligne. Chaque ligne commence par l'adresse début de ligne.

Les octets imprimés représentent le contenu de la mémoire en codes hexadécimaux.

Chaque ligne commence à une adresse modulo 16. En tête de liste, les octets non demandés sont remplacés par des points. La commande prend fin avec la frappe d'un caractère retour-chariot.

La commande M :           -M:FAC1 {  
-----                            } }  
                                  ,

. Visualisation et modification d'octets en mémoire.

Le premier octet à modifier est à l'adresse FAC1. Cette adresse doit être suivie d'un espace ou d'une virgule.

∅   visualisation de l'octet avant modification éventuelle  
,   modification de l'octet sans visualisation.

La modification d'un octet se fait en frappant une valeur hexadécimale ; seuls les 2 derniers caractères sont retenus. La modification est effective sur frappe d'un caractère blanc, virgule ou retour-chariot.

Un caractère blanc ou virgule provoque un passage à l'octet suivant (adresse suivante)

Le caractère retour-chariot termine la commande.



La commande Z :                -Z:FAC1,FAC2,FAC3

-----

. Initialisation d'une zone mémoire, par une valeur donnée par l'utilisateur

La zone comprise entre FAC1 et FAC2 est initialisée par la valeur du facteur FAC3, de la façon suivante :

FAC3 représente une valeur hexadécimale double octet. L'octet poids faibles représente la valeur initiale et l'octet poids forts est l'incrément de cette valeur.

Exemple :

FAC3 = 0	remise à zéro de la zone
FAC3 = 100	00, 01, 02, 03..... dans la zone (pas + 1)
FAC3 = FF00	00, FF, FE, FD..... dans la zone (pas - 1)

La commande T :                -T:FAC1, FAC2,FAC3

-----

. Contrôle de l'initialisation d'une zone mémoire.

Cette commande est le complément de la commande Z.

Elle permet de contrôler l'initialisation d'une zone.

La syntaxe est identique à celle de la commande Z.

Les octets non conformes sont édités sous la forme :

ADR    VA    VL ?

avec    ADR : adresse mémoire

        VA : valeur attendue

        VL : valeur lue

La commande \* :        - \* : [ FAC1 [ ,FAC2 ] ]  
-----

. Reprise de l'exécution d'un programme, à partir de l'adresse du dernier piège trouvé, avec ou sans pose d'un ou deux nouveaux pièges.

Un piège ne peut pas être placé sur l'adresse de reprise (→ pas d'instruction exécutée).

La commande ¥ :        - ¥ :  
-----

. Exécution d'un programme instruction par instruction

Le mode pas à pas est actif à partir du dernier piège rencontré. Chaque action sur la barre espace provoque l'exécution de l'instruction à l'adresse en cours, avec arrêt sur le prochain code opération.

A chaque instruction, l'état des registres et des indicateurs est affiché (voir commande G:)

Le mode pas à pas conserve le mode d'adressage (direct ou relatif)

La commande X :        - X :  
-----

. Changement du mode d'adressage (direct→relatif)

Cette commande permet lorsqu'on a rencontré un piège, de changer le mode d'adressage pour visualiser l'adresse relative (resp. directe) qui correspond à une adresse directe (resp. relative).

L'état des registres est à nouveau visualisé.

Deux commandes 'X' : successives permettant de revenir au mode d'adressage initial.

Remarque : Dans le cas où l'adresse du dernier piège rencontré est inférieure à l'adresse de base, seul l'adressage direct est possible. La commande X est alors inefficace.

# COMMANDES MOMIC / Z

=====

- A : visualisation-modification du 1er jeu de registres et indicateurs
- B : visualisation-modification du 2em jeu de registres et indicateurs
- S : visualisation-modification du pointeur de stack  
des registres d'index et du vecteur interruption
- K : visualisation-modification de l'octet option d'éditoin, de l'état  
de l'horloge et de l'état de masquage des niveaux d'interruption
- D : listage d'une zone mémoire
- M : visualisation-modification d'octets en mémoire
- & : définition de l'adresse d'implantation d'un programme
- G : exécution d'un programme avec/ sans 1 ou 2 pièges
- \* : reprise exécution après arrêt sur piège
- Ø : exécution d'un programme en pas à pas
- X : inversion du mode (ABS ou REL) d'exécution en pas à pas
- Z : initialisation d'une zone mémoire
- T : contrôle de l'initialisation mémoire.

INTRODUCTION

Le N G F - NOYAU DE GESTION DE FICHIERS - est un module regroupant les fonctions élémentaires d'une gestion de fichiers sur supports sectorisés.

Il offre à l'utilisateur, à partir d'une gestion dynamique de l'espace disponible sur son support et d'un accès direct aux enregistrements, la possibilité de :

- définir le(s) type(s) d'organisation de ses fichiers,
- construire la (les) méthode(s) d'accès à ses enregistrements.

Dans tout ce qui suit, la notion de secteurs de 256 octets sera utilisée.

La notion d'espace disque sera étendue aussi bien au disque dur qu'au disque souple.



## ORGANISATION DU CATALOGUE D'UN VOLUME SOUS NGF

Le catalogue d'un volume géré par le NGF débute sur le 5è secteur physique du volume (i.e. secteur n° 4).

Il comporte, en séquence alternée (cf. FIGURES 1 et 1-bis) :

- Une liste de 32 noms de fichiers (8 octets chacun) occupant la taille d'un secteur.
- Une liste des 32 descripteurs disque (64 octets chacun) respectifs à ces fichiers. Cette liste occupe 8 secteurs disque.
- Une nouvelle liste de 32 noms de fichiers .. etc..

Le nombre maximum de fichiers gérés sur un volume est limité à 255 (Fichiers numérotés de 1 à 255).

Le nom du fichier 0 (les 8 premiers octets du catalogue) est donc remplacé par un entête DESCRIPTEUR DE VOLUME.

Ce descripteur de volume, dont la création est à la charge de l'utilisateur, permet au NGF de reconnaître les caractéristiques du volume à gérer.

DESCRIPTEUR DE VOLUME (cf. FIGURE 1)

Ce descripteur occupe les 8 premiers octets du catalogue et permet de définir :

a) - La taille des granules sur le volume (LGR) :

Cette taille est exprimée en secteurs de 256 octets et ne peut prendre que les valeurs suivantes :

1 - 2 - 4 - 8 - 16 - 32 - 64 - 128 - 0 (c.a.d 256)

b) - La taille du catalogue (LCAT) :

Cette taille, exprimée en secteurs, représente le n° du premier secteur physique n'appartenant pas au catalogue. Le nombre maximal de fichiers sur un volume géré par le NGF étant limité à 255, la taille du catalogue est donc comprise entre les valeurs 13 et 76 (cf. FIGURE 1-bis)

c) - L'indicatif d'identification du catalogue (IND) :

Cet indicatif permet au NGF de reconnaître un volume géré par lui-même. Cet indicatif est défini égal à la valeur hexadécimale /9F.

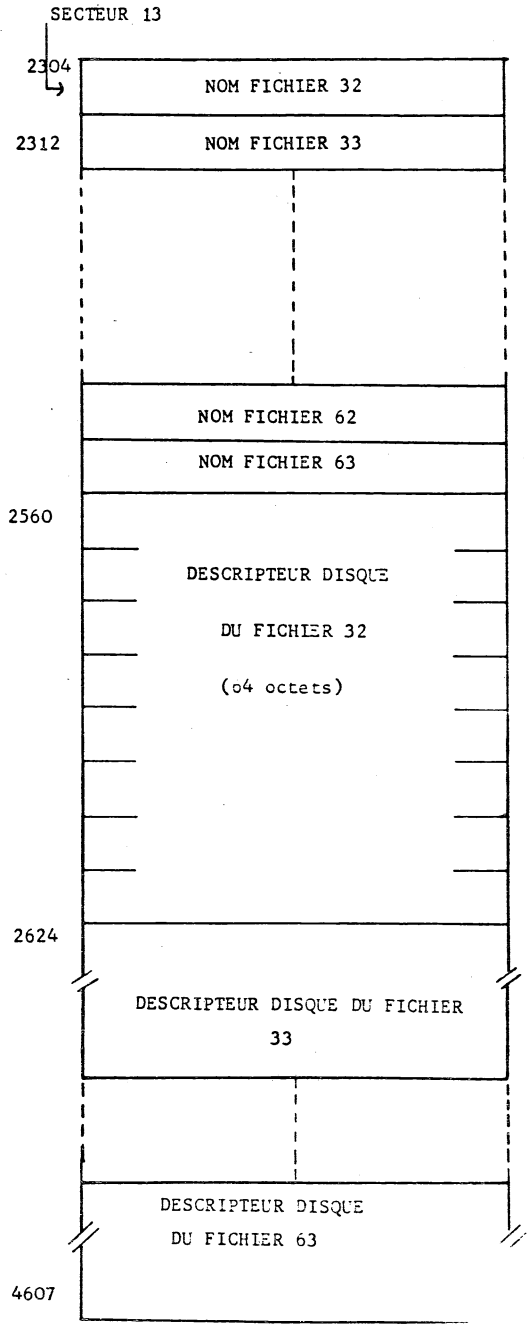
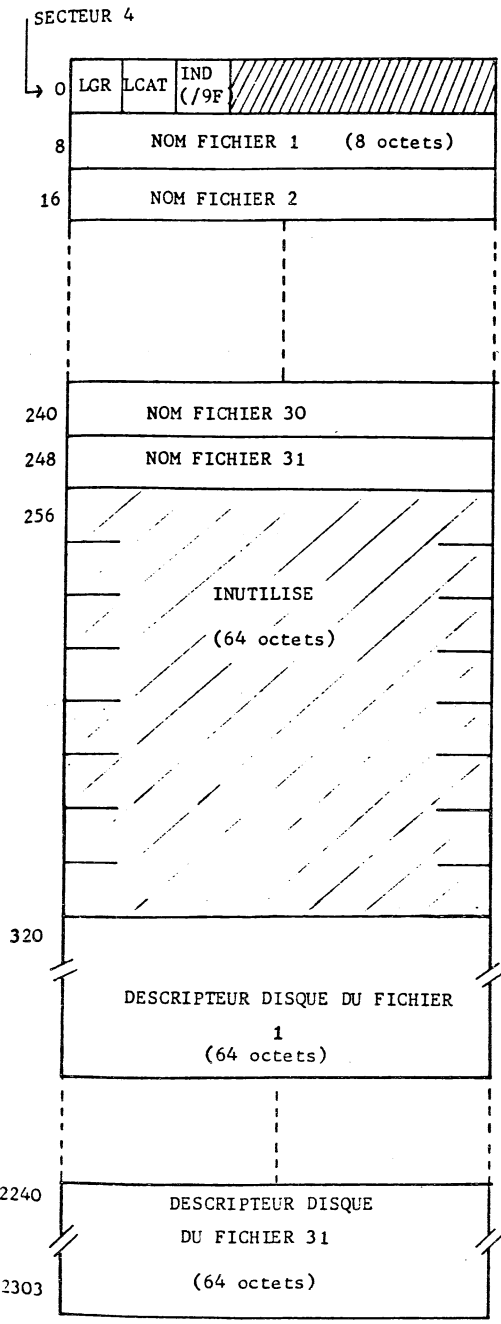


FIGURE 1

ORGANISATION DISQUE DU CATALOGUE

SECTEURS	INFORMATIONS
0	Table d'allocation du volume
1 à 3	Libre
4	Descripteur du volume + Noms fichiers 1 à 31
5 à 12	Descripteurs disque des fichiers 1 à 31
<u>13</u>	Noms Fichiers 32 à 63
14 à 21	Descripteurs disque des Fichiers 32 à 63
<u>22</u>	Noms Fichiers 64 à 95
23 à 30	Descripteurs disque des fichiers 64 à 95
⋮	⋮
⋮	⋮
⋮	⋮
<u>58</u>	Noms Fichiers 192 à 223
59 à 66	Descripteurs disque des Fichiers 192 à 223
<u>67</u>	Noms Fichiers 224 à 255
68 à 75	Descripteurs disque des Fichiers 224 à 255
<u>76</u>	

REMARQUE : Les valeurs soulignées représentent les longueurs possibles (LCAT) du catalogue.

FIGURE 1.bis

Le NGF affranchit l'utilisateur des problèmes d'estimation préalable de l'espace disque nécessaire à ses fichiers.

Il alloue en effet à chaque fichier, au fur et à mesure de ses besoins, un quantum d'espace disque appelé 'GRANULE'.

Un granule est un multiple de secteurs disque, défini à l'avance par l'utilisateur, en fonction de la capacité du support utilisé, et de la nature des fichiers à gérer (taille, taux d'expansion).

Ce paramètre (LGR), fixé par l'utilisateur au moment de la création du DESCRIPTEUR DE VOLUME, lui permet de rechercher une certaine optimisation dans la gestion de son espace disque.

- Il est à noter que le paramètre LGR, une fois défini, est généralisé à l'ensemble des fichiers du volume.

## TABLE D'ALLOCATION D'UN VOLUME SOUS NGF

Pour mémoriser l'allocation de l'espace disque à l'ensemble des fichiers d'un volume, le NGF dispose en début du volume (PISTE 0, secteur 0), d'une table représentant sur 254 x 8 bits (un bit à 0 représente un granule disponible) la capacité totale du volume.

Ceci permet, à titre d'exemple, avec des granules de 256 secteurs, de gérer jusqu'à 133. 169. 152 octets.

La table d'allocation du volume, gérée en simultanéité avec le descripteur disque d'un fichier, permet au NGF de suivre l'évolution de ce fichier sur le volume.

## DESCRIPTEUR DISQUE D'UN FICHIER (cf. FIGURE 2)

Ce descripteur mémorise toutes les informations particulières à un fichier (clés d'accès), ainsi que ses coordonnées successives pendant son évolution sur l'espace disque (blocs d'extension).

Le nombre maximum de blocs d'extension est limité à 19. Chaque bloc d'extension comporte les informations suivantes :

- Le nombre de granules physiquement contigus qui composent ce bloc (maximum = 255 granules par bloc)
- Le n° d'ordre du granule physique qui débute ce bloc.

Ces blocs d'extension permettent au NCF de chaîner l'espace disque alloué au fichier.

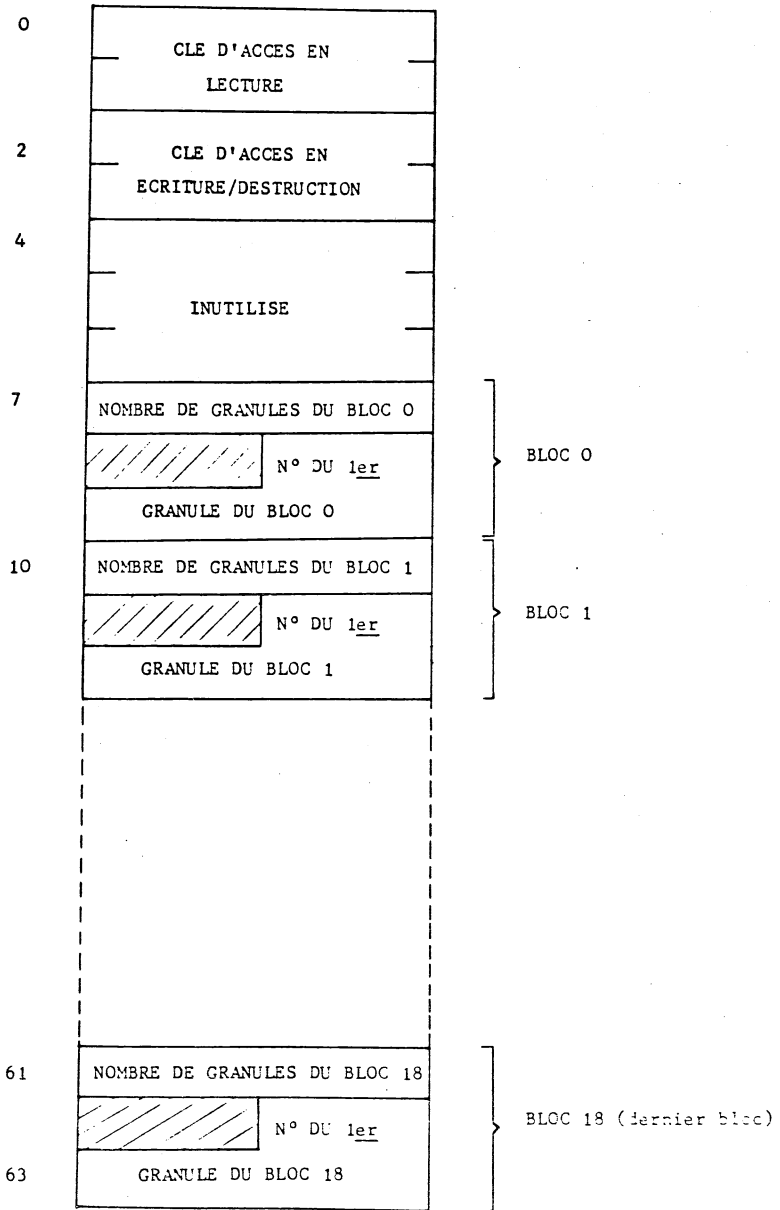


FIGURE 2



DESCRIPTEUR MEMOIRE D'UN FICHIER (64 octets)

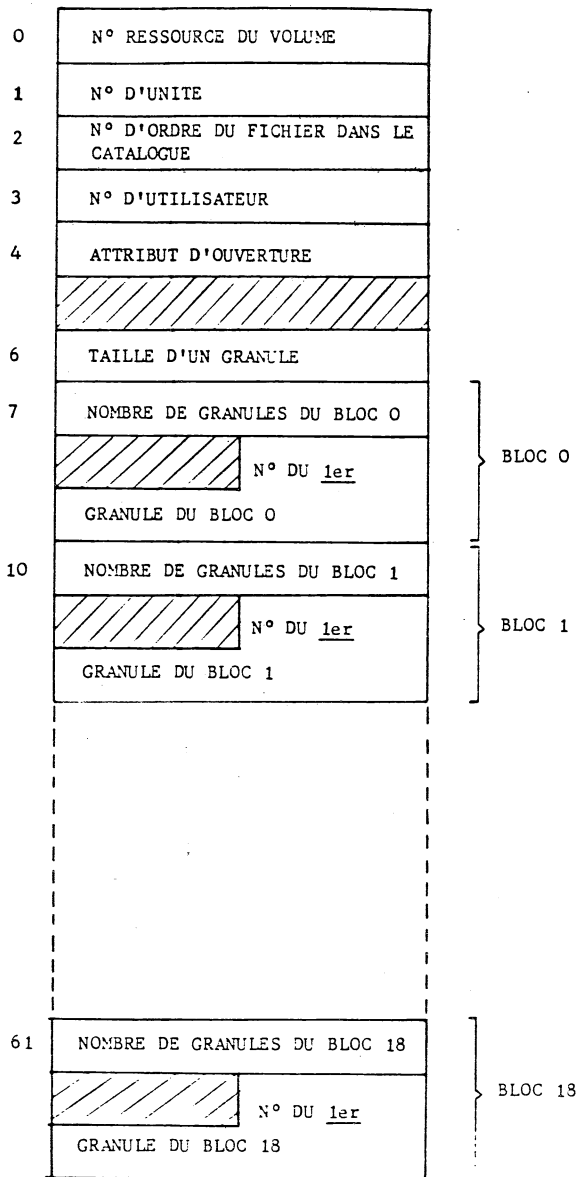


FIGURE 3

## TABLE DES NUMEROS LOGIQUES DU NGF (TNLG)

La longueur de cette table (i.e. le nombre de ses entrées) est un paramètre d'assemblage du NGF. Chaque entrée de TNLG occupe 64 octets mémoire.

Lorsqu'un fichier est activé, les informations relatives à son activation (i.e. les paramètres fournis par l'utilisateur dans son descripteur d'appel ) ainsi que celles relatives à son évolution sur le disque (blocs d'extension) sont dupliquées dans une entrée de TNLG, cette entrée constituant le descripteur mémoire du fichier (cf. FIGURE 3).

Le nombre d'entrées dans TNLG permet donc de traiter autant de fichiers en simultanéité :

PARAMETRAGE DU NGF :

Le NGF occupe 3K½ d'octets en mémoire (variables comprises).  
Il nécessite cependant une zone mémoire pour sa table des numéros logiques (TNLG), à raison de 64 octets par numéro logique.

Le nombre d'entrées dans TNLG est un paramètre d'assemblage de NGF. Dans la version de base, avec 16 numéros logiques, NGF occupe 4K½ d'octets mémoire.



## DEFINITION DES SUPPORTS SYSTEME ET UTILISATEUR

95H	numéro de ressource du support système par défaut
96H	numéro d'unité
97H	numéro de ressource du support utilisateur par défaut
98H	numéro d'unité

### ADRESSE DE CHARGEMENT

99H	adresse mémoire de chargement par défaut des overlays utilisateurs et système. (notation inversée).
-----	---

### TABLE DES PERIPHERIQUES

6 octets sont nécessaires pour décrire un périphérique.

- son mnémonique d'appel (2 octets)
- son nombre d'unités (1 octet)
- son numéro de ressource (1 octet)
- son adresse d'appel (2 octets) - (notation inversée)

COH	début de la table, 1er périphérique
C6H	2èm périphérique
etc...	



0002 0000  
 0004 0000  
 0006 0000  
 0008 0000  
 000A 0000  
 000C 0000  
 000E 0000

A DEFWD 0  
 A DEFWD 0  
 A DEFWD 0  
 A DEFWD 0  
 A DEFWD 0  
 A DEFWD 0  
 A DEFWD 0

TABLE DES PERIPHERIQUES

0000 464C  
 0002 02  
 0003 E0  
 0004 0001

A TPER  
 A  
 A  
 X

DEFM 'FL'  
 DEFB 2  
 DEFB FLD2  
 DEFW FLD2.

NOM DE SUPPORT  
 NB UNITES  
 RESSOURCE FLOPPY-D2  
 AER. IMPLANTATION

0006 4453  
 0008 02  
 0009 E2  
 000A 0002

A  
 A  
 A  
 X

DEFM 'DS'  
 DEFB 2  
 DEFB DSHB  
 DEFW DSHB.

NOM DE SUPPORT  
 NB UNITES  
 RESSOURCE DISQUE D120  
 AER. IMPLANTATION

000C 4653  
 000E 02  
 000F E1  
 0010 0003

A  
 A  
 A  
 X

DEFM 'FS'  
 DEFB 2  
 DEFB FLD1  
 DEFW FLD1.

NOM DE SUPPORT  
 NB UNITES  
 RESSOURCE FLOPPY SIMPLE DENSITE  
 AER. IMPLANTATION

0012 2020  
 0014 00  
 0015 00  
 0016 0000

A  
 A  
 A  
 A

DEFM '  
 DEFB 0  
 DEFB 0  
 DEFW 0

NOM DE SUPPORT  
 NB UNITES  
 NO RESSOURCE  
 AER. IMPLANTATION

0018 2020  
 001A 00  
 001B 00  
 001C 0000

A  
 A  
 A  
 A

DEFM '  
 DEFB 0  
 DEFB 0  
 DEFW 0

NOM DE SUPPORT  
 NB UNITES  
 NO RESSOURCE  
 AER. IMPLANTATION

001E 2020  
 0010 00  
 00E1 00  
 00E2 0000

A  
 A  
 A  
 A

DEFM '  
 DEFB 0  
 DEFB 0  
 DEFW 0

NOM DE SUPPORT  
 NB UNITES  
 NO RESSOURCE  
 AER. IMPLANTATION

0014 2020  
 00E6 00  
 00E7 00  
 00E8 0000

A  
 A  
 A  
 A

DEFM '  
 DEFB 0  
 DEFB 0  
 DEFW 0

NOM DE SUPPORT  
 NB UNITES  
 NO RESSOURCE  
 AER. IMPLANTATION

00EA 2020  
 00EC 00  
 00ED 00  
 00EE 0000

A  
 A  
 A  
 A

DEFM '  
 DEFB 0  
 DEFB 0  
 DEFW 0

NOM DE SUPPORT  
 NB UNITES  
 NO RESSOURCE  
 AER. IMPLANTATION

Z80 - 8080\*

## INSTRUCTIONS CLASSEES PAR MNEMONIQUES

8E	ADC (HL)	•	CB47	BIT 0,A	CB69	BIT 5,C
DD8E..	ADC (IX+d)		CB40	BIT 0,B	CB6A	BIT 5,D
FD8E..	ADC (IY+d)		CB41	BIT 0,C	CB6B	BIT 5,E
8F	ADC A	•	CB42	BIT 0,D	CB6C	BIT 5,H
88	ADC B	•	CB43	BIT 0,E	CB6D	BIT 5,L
89	ADC C	•	CB44	BIT 0,H	CB76	BIT 6,(HL)
8A	ADC D	•	CB45	BIT 0,L	DDCB..76	BIT 6,(IX+d)
8B	ADC E	•	CB4E	BIT 1,(HL)	FDCB..76	BIT 6,(IY+d)
8C	ADC H	•	DDCB..4E	BIT 1,(IX+d)	CB77	BIT 6,A
8D	ADC L	•	FDCB..4E	BIT 1,(IY+d)	CB70	BIT 6,B
CE..	ADC n	•	CB4F	BIT 1,A	CB71	BIT 6,B
ED4A	ADC HL,BC		CB48	BIT 1,B	CB72	BIT 6,D
ED5A	ADC HL,DE		CB49	BIT 1,C	CB73	BIT 6,E
ED6A	ADC HL,HL		CB4A	BIT 1,D	CB74	BIT 6,H
ED7A	ADC HL,SP		CB4B	BIT 1,E	CB75	BIT 6,L
86	ADD (HL)	•	CB4C	BIT 1,H	CB7E	BIT 7,(HL)
DD86..	ADD (IX+d)		CB4D	BIT 1,L	DDCB..7E	BIT 7,(IX+d)
FD86..	ADD (IY+d)		CB56	BIT 2 (HL)	FDCB..7E	BIT 7,(IY+d)
87	ADD A	•	DDCB..56	BIT 2,(IX+d)	CB7F	BIT 7,A
80	ADD B	•	FDCB..56	BIT 2,(IY+d)	CB78	BIT 7,B
81	ADD C	•	CB57	BIT 2,A	CB79	BIT 7,C
82	ADD D	•	CB50	BIT 2,B	CB7A	BIT 7,D
83	ADD E	•	CB51	BIT 2,C	CB7B	BIT 7,E
84	ADD H	•	CB52	BIT 2,D	CB7C	BIT 7,H
85	ADD L	•	CB53	BIT 2,E	CB7D	BIT 7,L
C6..	ADD n	•	CB54	BIT 2,H	DC....	CALL C,nn •
09	ADD HL,BC		CB55	BIT 2,L	FC....	CALL M,nn •
19	ADD HL,DE		CB5E	BIT 3,(HL)	D4....	CALL NC,nn •
29	ADD HL,HL		DDCB..5E	BIT 3,(IX+d)	CD....	CALL nn •
39	ADD HL,SP		FDCB..5E	BIT 3,(IY+d)	C4....	CALL NZ,nn •
DD09	ADD IX,BC		CB5F	BIT 3,A	F4....	CALL P,nn •
DD19	ADD IX,DE		CB58	BIT 3,B	EC....	CALL PE,nn •
DD29	ADD IX,IX		CB59	BIT 3,C	E4....	CALL PO,nn •
DD39	ADD IX,SP		CB5A	BIT 3,D	CC....	CALL Z,nn •
FD09	ADD IY,BC		CB58	BIT 3,E	3F....	CCF •
FD19	ADD IY,DE		CB5C	BIT 3,H	BE....	CP (HL) •
FD29	ADD IY,IY		CB5D	BIT 3,L	DDBE..	CP (IX+d)
FD39	ADD IY,SP		CB66	BIT 4,(HL)	FDBE..	CP (IY+d)
A6	AND (HL)	•	DDCB..66	BIT 4,(IX+d)	BF	CP A •
DDA6..	AND (IX+d)		FDCB..66	BIT 4,(IY+d)	B8	CP B •
FDA6..	AND (IY+d)		CB67	BIT 4,A	B9	CP C •
A7	AND A	•	CB60	BIT 4,B	BA	CP D •
A0	AND B	•	CB61	BIT 4,C	BB	CP E •
A1	AND C	•	CB62	BIT 4,D	BC	CP H •
A2	AND D	•	CB63	BIT 4,E	BD	CP L •
A3	AND E	•	CB64	BIT 4,H	FE..	CP n •
A4	AND H	•	CB65	BIT 4,L	EDA9	CPD
A5	AND L	•	CB6E	BIT 5,(HL)	EDB9	CPDR
E6..	AND n	•	DDCB..6E	BIT 5,(IX+d)	EDA1	CPI
C346	BIT 0,(HL)		FDCB..6E	BIT 5,(IY+d)	EDB1	CPIR
DDCB..46	BIT 0,(IX+d)		CB6F	BIT 5,A	2F	CPL
FDCB..46	BIT 0,(IY+d)		CB68	BIT 5,B	27	DAA •



35	DEC (HL)	•	DA....	JP C,nn	•	7D	LD A,L	•
DD35..	DEC (IX+d)		FA....	JP M,nn	•	3E..	LD A,n	•
FD35..	DEC (IY+d)		D2....	JP NC,nn	•	46	LD B,(HL)	•
3D	DEC A	•	C3....	JP nn	•	DD46..	LD B,(IX+d)	•
05	DEC B	•	C2....	JP NZ,nn	•	FD46..	LD B,(IY+d)	•
0B	DEC BC	•	F2....	JP P,nn	•	47	LD B,A	•
0D	DEC C	•	EA....	JP PE,nn	•	40	LD B,B	•
15	DEC D	•	E2....	JP PO,nn	•	41	LD B,C	•
1B	DEC DE	•	CA....	JP Z,nn	•	42	LD B,D	•
1D	DEC E	•	38..	JR C,e		43	LD B,E	•
25	DEC H	•	18..	JR e		44	LD B,H	•
2B	DEC HL	•	30..	JR NC,e		+5	LD B,L	•
DD2B	DEC IX		20..	JR NC,e		06..	LD B,n	•
FD2B	DEC IY		28..	JR Z,e		ED4B....	LD BC,(nn)	•
2D	DEC L	•	02	LD (BC),A	•	01....	LD B,nn	•
3B	DEC SP	•	12	LD (DE),A	•	4E	LD C,(HL)	•
F3	DI	•	77	LD (HL),A	•	DD4E..	LD C,(IX+d)	•
10..	DJNZ e		70	LD (HL),B	•	FD4E..	LD C,(IY+d)	•
FB	EI	•	71	LD (HL),C	•	4F	LD C,A	•
E3	EX (SP),HL	•	72	LD (HL),D	•	48	LD C,B	•
DDE3	EX (SP),IX		73	LD (HL),E	•	49	LD C,C	•
FDE3	EX (SP),IY		74	LD (HL),H	•	4A	LD C,D	•
08	EX AF,AF'		75	LD (HL),L	•	4B	LD C,E	•
EB	EX DE,HL	•	36..	LD (HL),n	•	4C	LD C,H	•
D9	EXX		DD77..	LD (IX+d),A		4D	LD C,L	•
76	HALT	•	DD70..	LD (IX+d),B		0E..	LD C,n	•
ED46	IMO		DD71..	LD (IX+d),C		56	LD D,(HL)	•
ED56	IMI		DD72..	LD (IX+d),D		DD56..	LD D,(IX+d)	•
ED5E	IM2		DD73..	LD (IX+d),E		FD56..	LD D,(IY+d)	•
ED78	IN A,(C)		DD74..	LD (IX+d),H		57	LD D,A	•
DB..	IN A,(n)	•	DD75..	LD (IX+d),L		50	LD D,B	•
ED40	IN B,(C)		DD36..	LD (IX+d),n		51	LD D,C	•
ED48	IN C,(C)		FD77..	LD (IY+d),A		52	LD D,D	•
ED50	IN D,(C)		FD70..	LD (IY+d),B		53	LD D,E	•
ED58	IN E,(C)		FD71..	LD (IY+d),C		54	LD D,H	•
ED60	IN H,(C)		FD72..	LD (IY+d),D		55	LD D,L	•
ED68	IN L,(C)		FD73..	LD (IY+d),E		16..	LD D,n	•
34	INC (HL)	•	FD74..	LD (IY+d),H		ED5B....	LD DE,(nn)	•
DD34..	INC (IX+d)		FD75..	LD (IY+d),L		11....	LD DE,nn	•
FD34..	INC (IY+d)		FD36..	LD (IY+d),n		5E	LD E,(HL)	•
3C	INC A	•	32....	LD (nn),A	•	DD5E..	LD E,(IX-d)	•
04	INC B	•	ED43....	LD (nn),BC		FD5E..	LD E,(IY+d)	•
03	INC BC	•	ED53....	LD (nn),DE		5F	LD E,A	•
0C	INC C	•	22....	LD (nn),HL	•	58	LD E,B	•
14	INC D	•	DD22....	LD (nn),IX		59	LD E,C	•
13	INC DE	•	FD22....	LD (nn),IY		5A	LD E,D	•
1C	INC E	•	ED73....	LD (nn),SP		5B	LD E,E	•
24	INC H	•	0A	LD A,(BC)	•	5C	LD E,H	•
23	INC HL	•	1A	LD A,(DE)	•	5D	LD E,L	•
DD23	INC IX		7E	LD A,(HL)	•	1E..	LD E,n	•
FD23	INC IY		DD7E	LD A,(IX+d)		66	LD H,(HL)	•
2C	INC L	•	FD7E	LD A,(IY+d)		DD66..	LD H,(IX+d)	•
33	INC SP	•	3A....	LD A,(nn)	•	FD66..	LD H,(IY-d)	•
EDAA	IND		7F	LD A,A	•	67	LD H,A	•
EDBA	INDR		78	LD A,B	•	60	LD H,B	•
EDA2	INI		79	LD A,C	•	61	LD H,C	•
EDB2	INIR		7A	LD A,D	•	62	LD H,D	•
E9	JP (HL)	•	7B	LD A,E	•	63	LD H,E	•
DDE9	JP (IX)		7C	LD A,H	•	64	LD H,H	•
FDE9	JP (IY)		ED57	LD A,I	•	65	LD H,L	•

26..	LD H,n	•	F5	PUSH AF	•	CBA8	RES 5,B	
2A....	LD HL,(nn)	•	C5	PUSH BC	•	CBA9	RES 5,C	
21....	LD HL,nn	•	D5	PUSH DE	•	CBAA	RES 5,D	
ED47	LD I,A		E5	PUSH HL	•	CBA8	RES 5,E	
DD2A....	LD IX,(nn)		DDE5	PUSH IX		CBAC	RES 5,H	
DD21....	LD IX,nn		FDE5	PUSH IY		CBAD	RES 5,L	
FD2A....	LD IY,(nn)		CB86	RES 0,(HL)		CB86	RES 6,(HL)	
FD21....	LD IY,nn		DDCB..86	RES 0,(IX+d)		DDCB..86	RES 6,(IX+d)	
6E	LD L,(HL)	•	FDCB..86	RES 0,(IY+d)		FDCB..86	RES 6,(IY+d)	
DD6E..	LD L,(IX+d)		CB87	RES 0,A		CB87	RES 6,A	
FD6E..	LD L,(IY+d)		CB80	RES 0,B		CB80	RES 6,B	
6F	LD L,A	•	CB81	RES 0,C		CB81	RES 6,C	
68	LD L,B	•	CB82	RES 0,D		CB82	RES 6,D	
69	LD L,C	•	CB83	RES 0,E		CB83	RES 6,E	
6A	LD L,D	•	CB84	RES 0,H		CB84	RES 6,H	
6B	LD L,E	•	CB85	RES 0,L		CB85	RES 6,L	
6C	LD L,H	•	CB8E	RES 1,(HL)		CB8E	RES 7,(HL)	
6D	LD L,L	•	DDCB..8E	RES 1,(IX+d)		DDCB..8E	RES 7,(IX+d)	
2E..	LD L,n	•	FDCB..8E	RES 1,(IY+d)		FDCB..8E	RES 7,(IY+d)	
ED4F	LD R,A		CB8F	RES 1,A		CB8F	RES 7,A	
ED7B	LD SP,(nn)		CB88	RES 1,B		CB88	RES 7,B	
F9	LD SP,HL	•	CB89	RES 1,C		CB89	RES 7,C	
DD79	LD SP,IX		CB8A	RES 1,D		CB8A	RES 7,D	
FD79	LD SP,IY		CB8B	RES 1,E		CB8B	RES 7,E	
31....	LD SP,nn	•	CB8C	RES 1,H		CB8C	RES 7,H	
EDA8	LDD		CB8D	RES 1,L		CB8D	RES 7,L	
EDB8	LDDR		CB96	RES 2,(HL)		CB96	RES 2,(HL)	
EDA0	LDI		DDCB..96	RES 2,(IX+d)		DDCB..96	RES 2,(IX+d)	
EDB0	LDIR		FDCB..96	RES 2,(IY+d)		FDCB..96	RES 2,(IY+d)	
ED44	NEG		CB97	RES 2,A		CB97	RES 2,A	
00	NOP	•	CB90	RES 2,B		CB90	RES 2,B	
B6	OR (HL)	•	CB91	RES 2,C		CB91	RES 2,C	
DDB6..	OR (IX+d)		CB92	RES 2,D		CB92	RES 2,D	
FDB6..	OR (IY+d)		CB93	RES 2,E		CB93	RES 2,E	
B7	OR A	•	CB94	RES 2,H		CB94	RES 2,H	
B0	OR B	•	CB95	RES 2,L		CB95	RES 2,L	
B1	OR C	•	CB9E	RES 3,(HL)		CB9E	RES 3,(HL)	
B2	OR D	•	DDCB..9E	RES 3,(IX+d)		DDCB..9E	RES 3,(IX+d)	
B3	OR E	•	FDCB..9E	RES 3,(IY+d)		FDCB..9E	RES 3,(IY+d)	
B4	OR H	•	CB9F	RES 3,A		CB9F	RES 3,A	
B5	OR L	•	CB98	RES 3,B		CB98	RES 3,B	
F6..	OR n	•	CB99	RES 3,C		CB99	RES 3,C	
EDBB	OTDR		CB9A	RES 3,D		CB9A	RES 3,D	
EDB3	OTIR		CB9B	RES 3,E		CB9B	RES 3,E	
ED79	OUT (C),A		CB9C	RES 3,H		CB9C	RES 3,H	
ED41	OUT (C),B		CB9D	RES 3,L		CB9D	RES 3,L	
ED49	OUT (C),C		CBA6	RES 4,(HL)		CBA6	RES 4,(HL)	
ED51	OUT (C),D		DDCB..A6	RES 4,(IX+d)		DDCB..A6	RES 4,(IX+d)	
ED59	OUT (C),E		FDCB..A6	RES 4,(IY+d)		FDCB..A6	RES 4,(IY+d)	
ED61	OUT (C),H		CBA7	RES 4,A		CBA7	RES 4,A	
ED69	OUT (C),L		CBA0	RES 4,B		CBA0	RES 4,B	
D3..	OUT (n),A	•	CBA1	RES 4,C		CBA1	RES 4,C	
EDAB	OUTD		CBA2	RES 4,D		CBA2	RES 4,D	
EDA3	OUTI		CBA3	RES 4,E		CBA3	RES 4,E	
F1	POP AF	•	CBA4	RES 4,H		CBA4	RES 4,H	
C1	POP BC	•	CBA5	RES 4,L		CBA5	RES 4,L	
D1	POP DE	•	CBAE	RES 5,(HL)		CBAE	RES 5,(HL)	
E1	POP HL	•	DDCB..AE	RES 5,(IX+d)		DDCB..AE	RES 5,(IX+d)	
DDE1	POP IX		FDCB..AE	RES 5,(IY+d)		FDCB..AE	RES 5,(IY+d)	
FDE1	POP IY		CBAF	RES 5,A		CBAF	RES 5,A	
						CBA8	RES 5,B	
						CBA9	RES 5,C	
						CBAA	RES 5,D	
						CBA8	RES 5,E	
						CBAC	RES 5,H	
						CBAD	RES 5,L	
						CB86	RES 6,(HL)	
						DDCB..86	RES 6,(IX+d)	
						FDCB..86	RES 6,(IY+d)	
						CB87	RES 6,A	
						CB80	RES 6,B	
						CB81	RES 6,C	
						CB82	RES 6,D	
						CB83	RES 6,E	
						CB84	RES 6,H	
						CB85	RES 6,L	
						CB8E	RES 7,(HL)	
						DDCB..8E	RES 7,(IX+d)	
						FDCB..8E	RES 7,(IY+d)	
						CB8F	RES 7,A	
						CB88	RES 7,B	
						CB89	RES 7,C	
						CB8A	RES 7,D	
						CB8B	RES 7,E	
						CB8C	RES 7,H	
						CB8D	RES 7,L	
						C9	RET	•
						D8	RET C	•
						F8	RET M	•
						DO	RET NC	•
						CO	RET NZ	•
						FO	RET P	•
						E8	RET PE	•
						EO	RET PO	•
						C8	RET Z	•
						ED4D	RETI	
						ED45	RETN	
						CB16	RL (HL)	
						DDCB..16	RL (IX+d)	
						FDCB..16	RL (IY+d)	
						CB17	RL A	
						CB10	RL B	
						CB11	RL C	
						CB12	RL D	
						CB13	RL E	
						CB14	RL H	
						CB15	RL L	
						17	RLA	•
						CB06	RLC (HL)	
						DDCB..06	RLC (IX+d)	
						FDCB..06	RLC (IY+d)	
						CB07	RLC A	
						CB00	RLC B	
						CB01	RLC C	
						CB02	RLC D	
						CB03	RLC E	
						CB04	RLC H	
						CB05	RLC L	
						07	RLCA	•
						ED6F	RLD	

CB1E	RR (HL)	CBCF	SET 1,A	CBFF	SET 7,A
DDCB..1E	RR (IX+d)	CBC8	SET 1,B	CBF8	SET 7,B
FDCB..1E	RR (IY+d)	CBC9	SET 1,C	CBF9	SET 7,C
CB1F	RR A	CBCA	SET 1,D	CBFA	SET 7,D
CB18	RR B	CBCB	SET 1,E	CBFB	SET 7,E
CB19	RR C	CBCC	SET 1,H	CBFC	SET 7,H
CB1A	RR D	CBCD	SET 1,L	CBFD	SET 7,L
CB1B	RR E	CBD6	SET 2,(HL)	CB26	SLA (HL)
CB1C	RR H	DECB..D6	SET 2,(IX+d)	DDCB..26	SLA (IX+d)
CB1D	RR L	FDCB..D6	SET 2,(IY+d)	FDCB..26	SLA (IY+d)
1F	RRA	CBD7	SET 2,A	CB27	SLA A
CBOE	RRC (HL)	CBDO	SET 2,B	CB20	SLA B
DDCB..OE	RRC (IX+d)	CBD1	SET 2,C	CB21	SLA C
FDCB..OE	RRC (IY+d)	CBD2	SET 2,D	CB22	SLA D
CBOF	RRC A	CBD3	SET 2,E	B23	SLA E
CBO8	RRC B	CBD4	SET 2,H	CB24	SLA H
CBO9	RRC C	CBD5	SET 2,L	CB25	SLA L
CBOA	RRC D	CBDE	SET 3,(HL)	CB2E	SRA (HL)
CBOB	RRC E	DDCB..DE	SET 3,(IX+d)	DDCB..2E	SRA (IX+d)
CBOC	RRC H	FDCB..DE	SET 3,(IY+d)	FDCB..2E	SRA (IY+d)
CBOD	RRC L	CBDF	SET 3,A	CB2F	SRA A
OF	RRCA	CBD8	SET 3,B	CB28	SRA B
ED67	RRD	CBD9	SET 3,C	CB29	SRA C
C7	RST 0	CBDA	SET 3,D	CB2A	SRA D
CF	RST 08H	CBD8	SET 3,E	CB2B	SRA E
D7	RST 10H	CBDC	SET 3,H	CB2C	SRA H
DF	RST 18H	CBDD	SET 3,L	CB2D	SRA L
E7	RST 20H	CBE6	SET 4,(HL)	CB3E	SRL (HL)
EF	RST 28H	DDCB..E6	SET 4,(IX+d)	DDCB..3E	SRL (IX+d)
F7	RST 30H	FDCB..E6	SET 4,(IY+d)	FDCB..3E	SRL (IY+d)
FF	RST 38H	CBE7	SET 4,A	CB3F	SRL A
9E	SBC (HL)	CBE0	SET 4,B	CB38	SRL B
DD9E..	SBC (IX+d)	CBE1	SET 4,C	CB39	SRL C
FD9E..	SBC (IY+d)	CBE2	SET 4,D	CB3A	SRL D
9F	SBC A	CBE3	SET 4,E	CB3B	SRL E
98	SBC B	CBE4	SET 4,H	CB3C	SRL H
99	SBC C	CBE5	SET 4,L	CB3D	SRL L
9A	SBC D	CBEE	SET 5,(HL)	96	SUB (HL)
9B	SBC E	DDCB..EE	SET 5,(IX+d)	DD96..	SUB (IX+d)
9C	SBC H	FDCB..EE	SET 5,(IY+d)	FD96..	SUB (IY+d)
9D	SBC L	CBEF	SET 5,A	97	SUB A
DE..	SBC n	CBE8	SET 5,B	90	SUB B
ED42	SBC HL,BC	CBE9	SET 5,C	91	SUB C
ED52	SBC HL,DE	CBEA	SET 5,D	92	SUB D
ED62	SBC HL,HL	CBEB	SET 5,E	93	SUB E
ED72	SBC HL,SP	CBEC	SET 5,H	94	SUB H
37	SCF	CBED	SET 5,L	95	SUB L
CBC6	SET 0,(HL)	CBF6	SET 6,(HL)	D6..	SUB n
DDCB..C6	SET 0,(IX+d)	DDCB..F6	SET 6,(IX+d)	AE	XOR (HL)
FDCB..C6	SET 0,(IY+d)	FDCB..F6	SET 6,(IY+d)	DDAE..	XOR (IX+d)
CBC7	SET 0,A	CBF7	SET 6,A	FDAE..	XOR (IY+d)
CBC0	SET 0,B	CBF0	SET 6,B	AF	XOR A
CBC1	SET 0,C	CBF1	SET 6,C	A8	XOR B
CBC2	SET 0,D	CBF2	SET 6,D	A9	XOR C
CBC3	SET 0,E	CBF3	SET 6,E	AA	XOR D
C3C4	SET 0,H	CBF4	SET 6,H	AB	XOR E
CBC5	SET 0,L	CBF5	SET 6,L	AC	XOR H
CBCE	SET 1,(HL)	CBEF	SET 7,(HL)	AD	XOR L
DDCB..CE	SET 1,(IX+d)	DDCB..FE	SET 7,(IX+d)	EE..	XOR n
FDCB..CE	SET 1,(IY+d)	FDCB..FE	SET 7,(IY+d)		



CONVERSIONS

- [NOMSUP.] CP, DV, NOMSUP1, NOMSUP2 [,1] v duplication volume
- [NOMSUP.] CP, CV, NOMSUP{,[LGGR] , [NBFIC] , [P]}v création volume
- [NOMSUP.] CP, CF, NOMFIC v création fichier
- [NOMSUP.] CP, SF, NOMFIC v suppression fichier
- [NOMSUP.] CP, RF, NOMFIC1, NOMFIC2 v renommer fichier
- [NOMSUP.] CP, DF, NOMFIC1, {[NOMSUP1.] [NOMFIC2]}v duplication fichier.

## LISTE DES CODES ERREUR

Erreurs périphérique

- 01 Unité non prête
- 02 Erreur lecture ou écriture
- 03 Erreur recherche piste
- 04 Paramètre incorrect dans un descripteur I'E/S dont :  
adresse secteur inexistante sur ce périphérique
- 05 Protection écriture
- 06 Erreur contrôleur périphérique
- 07 Erreur de synchronisation sur le secteur adressé
- 10 Erreur dans le secteur de remplacement
- 11 Contrôle écriture incorrect
- 12 Table des secteurs invalides incorrecte
- 13 Saturation de la table des secteurs invalides
- 14 Trop de secteurs invalides sur une piste
- 15 Formatage du secteur détruit.

Erreurs interpréteur de commandes

- 30 Syntaxe nom de support ou fichier incorrect
- 31 Numéro de périphérique inconnu
- 32 Unité non gérée par le système
- 33 Fichier de type incorrect
- 34 Syntaxe adresse de chargement incorrecte
- 35 Externe non résolu au chargement
- 36 Enregistrement incorrect dans un fichier objet
- 37 Défaut mémoire au chargement.

## Erreurs gestion de fichiers

- 40 Fichier inexistant
- 41 Fichier déjà existant
- 42 Fichier fermé
- 43 Fichier déjà ouvert
- 44 Réserve pour extension
- 45 Fichier comportant trop de blocs d'extension
- 46 Débordement du volume
- 47 Clés d'accès incorrectes
- 48 Fin de fichier
- 49 Catalogue saturé
- 50 Numéro logique incorrect
- 51 Table des numéros logiques saturée
- 52 Réserve pour extension
- 53 Adresse disque inconnue dans un fichier
- 54 Volume non créé par le système de fichiers.



