

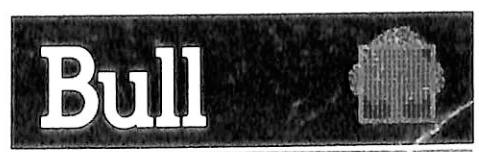
MICRAL

SYSTEME D'EXPLOITATION

PROLOGUE

Manuel de L'utilisateur

R-1003F



SYSTEME D'EXPLOITATION PROLOGUE DE R2E
MANUEL DE L'UTILISATEUR

Publication No. R-1003F

Edition Février 1983

SYSTEME D'EXPLOITATION SUR DISQUE PROLOGUE
MANUEL DE L'UTILISATEUR

TABLE DES MATIERES

<u>Paragraphe</u>	<u>Titre</u>	<u>Page</u>
<u>CHAPITRE 1. INTRODUCTION</u>		
1.1	Description Générale	1-1
1.2	Les Possibilités de PROLOGUE	1-1
1.2.1	Les Programmes du Noyau de PROLOGUE	1-1
1.2.2	Programmes optionnels de Traitement de Langage de haut niveau	1-2
1.2.3	Programmes de Développement de Système PROLOGUE	1-2
1.2.4	Interfaçage avec le Système PROLOGUE	1-3
1.2.5	Configuration d'un système PROLOGUE	1-3
1.3	Organisation du Système	1-4
1.4	Organisation des Fichiers	1-5
1.5	Structure des Commandes	1-6
1.5.1	Conventions de Notation	1-6
1.5.2	Références au Fichier -- Nom de Fichier	1-7
1.5.3	Références de Fichiers -- Nom de Programme	1-11
1.5.4	Chargement de PROLOGUE	1-11
1.5.5	Syntaxe Générale des Commandes PROLOGUE	1-12
1.6	Détection d'erreur	1-15
<u>CHAPITRE 2. COMMANDES ET ROUTINES STANDARD DE PROLOGUE</u>		
2.1	Introduction	2-1
2.2	Programmes du noyau de PROLOGUE	2-1
2.3	Répertoire	2-2
2.3.1	Répertoire d'un volume	2-2
2.3.1.1	Format d'affichage	2-3
2.3.2	Fichiers Sélectionnés dans le Répertoire	2-5
2.4	Editeur de Texte	2-6
2.4.1	Généralités	2-6
2.4.2	Syntaxe	2-7
2.4.3	Utilisation de l'Editeur	2-8
2.4.4	Commandes de l'Editeur	2-9
2.4.4.1	Commandes pour déplacer le curseur	2-9
2.4.4.2	Commandes Insertion/Modification/Suppression	2-10
2.4.4.3	Commandes de Recherche de Chaîne	2-11
2.4.4.4	Fin des Commandes de Mise à Jour	2-13
2.4.4.5	Commande de Longueur de Ligne	2-13
2.5	Utilitaire CP (Copie)	2-13
2.5.1	Création de Volume	2-14
2.5.2	Création d'un Fichier	2-16
2.5.3	Suppression d'un Fichier	2-16
2.5.4	Changement de Nom d'un Fichier	2-17
2.5.5	Reproduire un Fichier	2-17
2.5.6	Reproduction d'un Volume	2-18
2.5.7	Changement de Nom d'un Volume	2-19
2.5.8	Désignation de groupe de fichiers pour les commandes de suppression et duplication	2-19

<u>Paragraphe</u>	<u>Titre</u>	<u>Page</u>
2.6.	CPS - Copie des Erreurs	2-20
2.6.1	Mode de Copie Totale	2-20
2.6.1.1	Description	2-20
2.6.1.2	Affichage d'erreur	2-23
2.6.2	Mode de Copie Libre	2-24
2.6.2.1	Description	2-24
2.6.2.2	Procédure	2-25
2.6.2.3	Affichage d'erreur	2-26
2.7	STATUS	2-27
2.8	PATCH	2-29
2.8.1	Description	2-29
2.8.2	Modification du Secteur	2-29
2.9	DATE	2-30
2.10	ASG - Assignation	2-32
2.10.1	Assignation du Fichier des Commandes	2-32
2.10.1.1	Format des Commandes	2-32
2.10.1.2	Format du Fichier de Commandes	2-33
2.10.1.3	Commentaires	2-34
2.11	TSM - Test des Disques Magnétiques	2-35
2.12	TELE - Procédure de Télétransmission	2-36
2.12.1	Configuration Matérielle Exigée par Le Système	2-36
2.12.2	Syntaxe de La Commande	2-36
2.12.3	Définition de La Transmission	2-37

CHAPITRE 3. LE BASIC RZE SOUS PROLOGUE

3.1	Généralités	3-1
3.2	Utilisation de l'Interpréteur BASIC sous PROLOGUE	3-1
3.2.1	Invocation de BASIC	3-1
3.2.2	Modes d'Opérations	3-2
3.2.3	Numéros de Lignes	3-3
3.2.4	Commandes de Programme/Fichier BASIC	3-3
3.2.4.1	Commande RUN	3-3
3.2.4.2	Commande SAVE	3-4
3.2.4.3	Commande LOAD	3-4
3.2.4.4	Commande MERGE	3-5
3.2.4.5	Commande KILL	3-5
3.2.4.6	Commande NAME	3-5
3.2.4.7	Commande FILES	3-5
3.2.4.8	Autres Commandes	3-6
3.2.5	Instructions Mettant en Jeu des Fichiers de Données	3-6
3.3	Utilisation du Compilateur BASIC Microsoft sous PROLOGUE	3-6
3.4	Utilisation de l'Editeur de Liens Microsoft L80 sous PROLOGUE	3-7

CHAPITRE 4. BAL SOUS PROLOGUE

4.1	Généralités	4-1
4.2	Le Traducteur BAL - TR	4-1
4.3	L'Exécuteur BAL - EX	4-6

<u>Paragraphe</u>	<u>Titre</u>	<u>Page</u>
	<u>CHAPITRE 5. FORTRAN-80 MICROSOFT SOUS PROLOGUE</u>	5-1
	<u>CHAPITRE 6. COBOL-80 MICROSOFT SOUS PROLOGUE</u>	6-1
	<u>CHAPITRE 7. ORGANISATION DE SYSTEME DE FICHIERS</u>	
7.1	Généralités	7-1
7.2	Noyau du Système de Gestion des Fichiers	7-1
7.3	Organisation de l'espace disque	7-1
7.3.1	Le Granule - L'Unité de Stockage Logique	7-2
7.3.2	Table d'Allocation de Mémoire Disque	7-2
7.3.3	Considérations importantes concernant l'espace	7-2
7.4	Paramètres du Système	7-3
	<u>ANNEXE A. CODES D'ERREUR DE PROLOGUE</u>	
A-1	Généralités	A-1
A-2	Codes d'Erreur de Périphériques	A-1
A-3	Codes d'Erreurs de Moniteur Temps Réel (MTZ)	A-4
A-4	Codes d'Erreur de l'Interpréteur de Commande	A-5
A-6	Codes Erreurs de l'Accès Séquentiel	A-8
A-7	Codes Erreurs de l'Accès Séquentiel Indexé	A-9
A-8	Erreurs d'exécution en BAL	A-10
	<u>ANNEXE B. BIBLIOGRAPHIE</u>	B-1
	<u>ANNEXE C. SOMMAIRE DES COMMANDES DE PROLOGUE</u>	C-1

CHAPITRE 1. INTRODUCTION

1.1 DESCRIPTION GENERALE

PROLOGUE est un système d'exploitation sur disque pour les microordinateurs RZE qui offre un environnement global pour la création, le stockage, la vérification et l'exécution de programmes. Il accepte des disques souples, des disques durs ou une combinaison des deux. PROLOGUE supporte une structure de fichiers identifiés par nom, ce qui permet une allocation dynamique d'espace de fichier. Ainsi, l'utilisateur crée et manipule des fichiers en donnant leur nom, tandis que le système se charge des détails de la gestion pour stocker de façon efficace les fichiers sur les disques.

1.2 LES POSSIBILITES DE PROLOGUE

PROLOGUE offre un grand nombre de fonctions système qu'on peut diviser en deux catégories comme suit :

1. Fonctions Utilisateur - Ce sont ces fonctions du système d'exploitation normalement utilisées pour la création et l'exécution des programmes d'application lorsqu'on travaille en langage de haut niveau.
2. Fonctions Système - Ce sont ces fonctions du système d'exploitation normalement utilisées seulement par les programmeurs système expérimentés, travaillant en langage assembleur.

1.2.1 Les Programmes du Noyau de PROLOGUE

PROLOGUE est organisé autour d'un noyau de routines standard pour la gestion des fichiers et le système d'exploitation qui sont fournis avec tous les systèmes PROLOGUE. Lorsque PROLOGUE est chargé, ces routines deviennent résidentes dans la partie basse de la mémoire.

A la demande de l'utilisateur, l'option Système de Gestion de Fichiers Séquentiels et/ou Séquentiels Indexés peut être configurée dans le système PROLOGUE. Cela exige un supplément d'espace dans la partie basse de la mémoire, c'est pourquoi ces possibilités ne sont incluses qu'en cas de besoin.

Un certain nombre de routines de support standard dans PROLOGUE sont fournies sur le disque PROLOGUE de l'utilisateur sous forme de modules objets qui sont chargés et exécutés en cas de besoin au lieu d'être résidents.

Ces modules sont :

- / - Routine Répertoire de Disque
- ED - Editeur de Texte -- Crée et modifie les fichiers source.

- CP - Utilitaire de Copie -- Crée et manipule les volumes et fichiers PROLOGUE.
- CPS - Utilitaire de Copie Secteur -- Copie tout ou partie d'un disque.
- STATUS - Affiche le statut opérationnel des périphériques magnétiques.
- PATCH - Permet la modification directe des données dans tout secteur d'un disque.
- DATE - Initialise l'heure et la date dans le système.
- ASG - Assignation -- Crée des fichiers de procédure qui permettent au système d'exécuter automatiquement une série de commandes PROLOGUE.
- TSM - Test Périphérique Magnétique -- Teste le fonctionnement correct des unités magnétiques.
- TELE - Télétransmission -- Transmet des fichiers entre deux sites d'ordinateurs par une interface de communication V24.

Ces programmes standard sont définis en détail au Chapitre 2 de ce document.

1.2.2 Programmes optionnels de Traitement de Langage de Haut Niveau

Différents Programmes de Traitement de Langage de Haut Niveau de RZE fonctionnent sous le contrôle du système d'exploitation PROLOGUE et sont fournis comme options. Ce sont :

- MBASIC -- Interpréteur BASIC Microsoft
- BASCOM -- Compilateur BASIC Microsoft
- L80 -- Editeur de Liens Microsoft
- BAL -- BASIC Etendu Orienté Gestion de RZE
- FORTRAN-80 -- FORTRAN Microsoft
- COBOL-80 -- COBOL Microsoft (futur produit)
- M80 -- Macro-assembleur Microsoft

L'appel de chacun de ces programmes est décrit aux Chapitres 3-6 de ce document. Les langages eux-mêmes sont décrits dans les manuels de référence de programmation particuliers. Voir Annexe B pour la liste des manuels appropriés.

1.2.3 Programmes de Développement de Système PROLOGUE

Une version Développement de Système de PROLOGUE est disponible et est utilisée principalement par les hommes système qui travaillent au niveau du langage assembleur. Elle contient toutes les routines de PROLOGUE de la version utilisateur, plus les modules suivants :

- AZM - Assembleur -- Traite les fichiers source en langage assembleur (mnémoniques Zilog) pour créer des programmes objets exécutables.

- MAC - Macro-Assembleur R2E -- Traite les fichiers sources en langage assembleur (mnémoniques et pseudo-codes d'opération Zilog) pour créer des programmes objets exécutables.
- EDL - Editeur de Liens -- Lie les modules objets assemblés en résolvant les références externes.
- MM - Moniteur -- Aide à la mise au point des programmes en langage assembleur permettant de lire et de modifier les registres de l'unité centrale et la mémoire.
- CVB - Convertisseur -- Convertit un programme objet en binaire.
- RELOC - Convertit un programme absolu en un programme translatable.
- DIO - Donne le contrôle pour la programmation d'une mémoire PROM d'E/S

L'utilisation de ces programmes est décrite en détail dans le Guide du Programmeur pour le Système d'Exploitation et la Gestion des Fichiers PROLOGUE, publication numéro R-1004.

1.2.4 Interfaçage avec le Système PROLOGUE

Le Système d'Exploitation PROLOGUE comporte différentes routines et fonctions écrites en Langage Assembleur. Ces routines sont utilisables par le programmeur dans des programmes écrits en Langage Assembleur et en d'autres langages qui possèdent la fonction d'appel à un sous-programme assembleur. Ces routines sont décrites en détail dans le Guide de Programmeur Système de PROLOGUE, publication numéro R-1004.

Ces routines sont:

- Routine d'entrée/sortie directe sur périphérique
- Routine d'entrée par console
- Routines sur console et sur imprimante de ligne
- Routines de création/ouverture/lecture/écriture/fermeture/suppression de fichiers
- Routine d'analyse de fichier
- Routine de chargement de fichier objet
- Routine de conversion ASCII/binaire.

1.2.5 Configuration d'un système PROLOGUE

Un système PROLOGUE (sur disque ou disquette) peut être configuré par une procédure normalement établie par un programmeur système. Dans le processus de configuration, l'opérateur peut modifier des paramètres du système et choisir des unités périphériques spécifiques et des options système. Cela permet de donner à chaque système d'exploitation PROLOGUE la configuration la plus efficace pour chaque installation d'ordinateur en ne retenant que les logiciels de contrôle périphériques et les routines nécessaires pour ce système.

Chaque ordinateur est livré avec un système PROLOGUE correctement configuré. Si des unités sont ajoutées ou changées, le programmeur-système peut avoir à reconfigurer le système. Se référer au Guide du Programmeur Système pour le Système d'exploitation et la Gestion des fichiers PROLOGUE (PROLOGUE Operating System and File Management System Programmer's Guide, publication numéro R-1004) pour les directives complètes sur l'utilisation du configurateur PROLOGUE.

Pour l'information générale de l'utilisateur, les options qu'on peut sélectionner dans PROLOGUE sont listées ci-dessous :

1. Paramètres modifiables du système

- * Adresse la plus élevée de mémoire utilisable (exemple, /FB80)
- * Mémoire la plus basse pour programmes utilisateur - Adresse de chargement (exemple, /270E)
- * Nom mnémorique pour les types d'unités (exemple, FL)
- * Unité standard pour les programmes système - Unité système (exemple, FLO)
- * Unité standard pour les programmes et fichiers utilisateur - Unité utilisateur (exemple, FL1)
- * Lignes et colonnes pour imprimantes
- * Date et heure initiales par défaut
- * Langue des messages
- * Marque décimale

2. Contrôleurs de périphérique possibles (écran et disque obligatoires, tous les autres optionnels)

- * Ecran - CDC, R2E type 8020, 8021, 8030, 8031
- * Imprimante - CENTRONICS pour 8020/8021, 8030/8031 ; QUME pour 8021, 8030
- * Disque souple - pour 8020/8021 ; pour 8030/8031, 8 pouces pour 8040 (secteurs fixes)
- * Disque dur - 5 pouces pour 8021 ; D120/D140 pour 8030/8031/8040
- * Bande magnétique
- * Autre (voir Guide du Programmeur PROLOGUE)

3. Sous-systèmes logiciel possibles (tous optionnels)

- * Méthode d'Accès Séquentiel (SAM)
- * Méthode d'Accès Séquentiel Indexé (ISAM)
- * Moniteur multi-tâches (MTZ)
- * Protocole de Communications (BSC)
- * Chaînage des commandes par assignation d'une entrée de commande à un fichier de commande

1.3 ORGANISATION DU SYSTEME

Pour utiliser de la façon la plus efficace PROLOGUE et les langages et routines qu'il contrôle, il est important d'avoir une compréhension générale de l'organisation des fichiers et de la structure des commandes utilisées par le système. Cette organisation est examinée ci-dessous avec assez de détails pour donner une compréhension générale. Des informations détaillées et complètes pour le programmeur système sont donnés dans le Guide du program-

1.4 ORGANISATION DES FICHIERS

Etant donné que PROLOGUE est un système d'exploitation sur disque, il doit assurer la gestion ordinaire qu'implique la maintenance d'un système de fichier identifié par nom, pour libérer l'utilisateur en lui permettant de spécifier et de manipuler tout fichier en le désignant par son nom, sans avoir à se préoccuper des détails physiques du stockage et de l'extraction des fichiers sur le disque.

Pour permettre à PROLOGUE de gérer convenablement les fichiers, l'utilisateur doit préciser certains aspects du système de fichier en introduisant les paramètres au moment de la création de ces fichiers. La terminologie utilisée est définie ci-dessous.

- Volume - Le système PROLOGUE est organisé par volume, où chaque support magnétique -- comme par exemple, une mini disquette souple -- est un volume possédant son propre nom et ses propres caractéristiques logiques de stockage. Chaque volume comporte un répertoire des fichiers qu'il contient. (Dans ce manuel, le terme disque est employé pour désigner un tel support magnétique.)
- Piste - Tous les supports disques magnétiques sont organisés en une série de pistes magnétiques concentriques. Par exemple, les mini disquettes souples utilisées sur plusieurs ordinateurs R2E ont 35 pistes par face.
- Secteur - Chaque disque est divisé en un certain nombre de sections radiales (secteurs), de façon telle que chaque piste est formée du même nombre de secteurs - 16 dans le cas des mini disquettes souples.
- Granule - C'est la plus petite unité logique de stockage sur un volume individuel du système PROLOGUE. Lorsqu'un volume est créé, le programmeur doit spécifier le nombre de secteurs contenus dans chaque granule. Chaque granule dans ce volume sera alors composé par le nombre indiqué de secteurs consécutifs sur le disque. Ce nombre est une puissance de 2 (1.2.4.8.16...). Le nombre minimum de secteurs par granule est 1 ; le maximum est 256. La valeur par défaut est 16, nombre de secteurs contenus dans une piste du mini disque. Etant donné le nombre limité maximum à enregistrer sur un disque, le nombre minimum de secteurs par granule sur un disque de 10 méga-octets doit être défini par le programmeur comme égal à 32, sinon la commande sera refusée.

Comme un seul fichier peut utiliser un granule donné, cette notion de taille de granule est importante.

Cette organisation permet au programmeur de spécifier les unités logiques en mémoire avec la taille qui convient le mieux à ses besoins.

- Bloc - Un bloc est formé de un ou plusieurs granules consécutifs.
- Fichier - Tous les programmes et toutes les données sont stockés sur disque sous forme de fichiers. Chaque fichier a un nom qui l'identifie uniquement et qui est formé d'un nom proprement dit, d'une désignation de type facultative et d'une clé d'accès au fichier facultative (voir paragraphe 1.5.2 pour plus de détails). Le système garde trace des fichiers avec le nombre de blocs et de granules.

- Unités de Support - Tous les supports magnétiques dans le système sont organisés en volumes logiques qui doivent être physiquement montés sur une unité périphérique magnétique comme, par exemple, une unité de disque souple ou une unité de disque dur. Dans les commandes de PROLOGUE, l'utilisateur peut spécifier l'unité sur laquelle le système doit localiser le volume ou le fichier spécifié. Si l'utilisateur omet cette spécification, le système suppose une situation par défaut comme suit :

Unité de Support Système -- C'est l'unité qui contient les programmes du système comme PROLOGUE lui-même, les utilitaires de répertoire, de copie, BASIC, etc. Cette unité est définie au moment de la configuration du système et est généralement l'unité 0.

Unité de Support Utilisateur -- C'est l'unité contenant les fichiers de l'utilisateur. Cette unité est également définie au moment de la configuration du système et est généralement l'unité 1.

Dans un système à base de disque souple, les unités par défaut seront les unités de disque souple 0 et 1. Dans un système à disque dur, les unités par défaut seront les unités de disque dur. L'utilisateur peut changer ces désignations par défaut en changeant certains des paramètres de mémoire comme on le verra au paragraphe 7.4.

Une fois l'information de base fournie, PROLOGUE gère automatiquement les fichiers en allouant automatiquement l'espace disque par granule en ajoutant et en supprimant les données pour utiliser de la façon la plus efficace l'espace disponible.

1.5 STRUCTURE DES COMMANDES

1.5.1 Conventions de Notation

L'opérateur communique avec PROLOGUE à l'aide d'un ensemble de

commandes définies avec précision. La structure générale de ces commandes est décrite dans ce qui suit. Les conventions de notation suivantes sont employées :

- { } - Indique un choix d'informations à entrer.
- [] - Indique une zone facultative dans une commande. Si l'information facultative n'est pas entrée, une valeur par défaut implicite est prise. (Noter que les crochets ne doivent pas être frappés lorsque la commande est introduite dans l'ordinateur.)

b - Indique un blanc obligatoire.

cr ou (cr) - Indique un retour de chariot. Normalement frappé pour terminer une ligne de commande.

1.5.2 Références au Fichier -- Nom de Fichier

PROLOGUE supporte un système de fichiers identifiés par nom, avec des fichiers multiples stockés, catalogués et disponibles sur un seul disque dur ou une seule disquette. Tous les fichiers de programmes et de données dans le système PROLOGUE sont identifiés par un nom de fichier qui a la structure suivante :

Nomfichier=[Unité.]Nom[-Type][:Clé]

Nom et numéro d'unités facultatifs, de l'unité où le fichier est physiquement localisé (par exemple, FL1).

Point obligatoire seulement lorsque Unité est spécifiée.

Nom de fichier de 1 à 7 caractères

Désignation de Type de fichier en un seul caractère. Doit être précédé d'un tiret.

Désignation de Clé d'Accès de 1 à 4 caractères. Doit être précédé de deux-points.

Où :

Unité - Nom et numéro d'unité de l'unité où le fichier est physiquement situé, comme par exemple :

FL0 - Disque souple (floppy disk), unité 0

FL1 - Disque souple (floppy disk), unité 1

- DS0 - Disque dur, unité 0
- DS1 - Disque dur, unité 1
- DS2 - Disque dur, unité 2, etc.

Cas par Défaut : Si l'unité n'est pas spécifiée, c'est l'unité implicite qui est choisie. Cette unité est définie à la configuration du système (voir discussion des paramètres au Paragraphe 7.5) et se trouve être habituellement l'unité 1. (L'unité de support du système à partir duquel PROLOGUE est automatiquement chargé en mémoire est en général l'unité 0).

- Nom** - Un nom de 1 à 7 caractères est obligatoire. Ces caractères peuvent être alphabétiques, numériques ou les caractères spéciaux \$ et /. Les autres caractères spéciaux doivent être évités. Le point, le tiret et les deux-points ne doivent pas être utilisés dans un nom.
- Type** - Un seul caractère facultatif, servant à désigner le type de fichier. Tout caractère (alphabétique, numérique ou \$) peut être utilisé, la liste qui suit donne cependant plusieurs désignations de types utilisés par les programmes du système et il est recommandé de s'y conformer :
- 0 - Objet. Fichier qui peut être chargé et exécuté par PROLOGUE. S'il est translatable, il peut recevoir une adresse de chargement fixe en mémoire. Les fichiers du noyau de PROLOGUE et les fichiers optionnels des processeurs de langage figureront dans le répertoire de votre système avec un type 0.
 - B - Binaire. Fichier binaire absolu qui peut être chargé dans une zone de mémoire spécifique par le chargeur "bootstrap" et être exécuté sans l'aide de PROLOGUE.
 - S - Source. Fichier source ASCII. Peut être créé par l'Editeur pour définir des lignes sources de langages (comme le BAL ou le FORTRAN) ou des commandes pour ASG (Assignment).

Les fichiers sources écrits sous l'interpréteur BASIC et stockés sur disque à l'aide de la commande SAVE (avec l'option A) recevront le type S comme valeur par défaut.

- R - Fichier compilé. Généré par des compilateurs (BASCOR, F80...) et chargeable par le chargeur L80.
- L - Liste. Fichier de listing de programme généré par les compilateurs.
- T - Code Intermédiaire Traduit. Généré par le traducteur BAL. Ne peut être chargé et exécuté que par le programme EX (Exécuteur) de BAL.
- I - Fichier de Clés en Séquentiel Indexé. Généré et utilisé par le système de Gestion des Fichiers Séquentiels Indexés.
- D - Fichier de Données. En correspondance avec le fichier de clés; généré et utilisé par le système de fichiers séquentiels indexés.
- S - Fichier Temporaire. Créé par des utilitaires PROLOGUE.

Cas par Défaut : Si la désignation du type est omise sur les fichiers que l'utilisateur crée avec le programme Copie (CP), un blanc ASCII est supposé. Si la désignation de type est omise lorsqu'on spécifie des fichiers à créer par certains programmes de PROLOGUE, une désignation appropriée sera assignée. Par exemple l'éditeur de liens n'acceptera que les fichiers de type D et assignera le type O au fichier de sortie.

Clé

- Clé d'Accès de Sécurité (mot de passe) de 1 à 4 caractères. La clé n'est affichée qu'à la première fois où elle est entrée. Elle n'est pas et ne peut pas être affichée avec le répertoire des fichiers.

Cette disposition est utile pour protéger certains programmes contre des modifications non autorisées car un fichier ne peut pas être écrit, modifié, copié ou supprimé sans que la clé correcte soit spécifiée dans le nom du fichier.

Le format de la clé est RRWW ; et il peut être utilisé comme suit :

Aucune clé spécifiée - blancs ASCII supposés (cas par défaut)

Clé = RR

- Clé stockée sous la forme RRbb. La frappe de RR comme clé pour le nom de fichier permet la lecture, l'écriture et la suppression.

CLÉ = RRWW

- L'entrée de la seule partie RR de la clé ne permet que la lecture. L'entrée de RRWW permet la lecture, l'écriture et la suppression.

Exemple

- Supposons que NEW:1234 est créé. L'entrée de NEW:12 permet la lecture du fichier, mais interdit l'écriture ou la suppression. L'entrée de NEW:1234 permet la lecture, l'écriture et la suppression.

Remarque : Ne pas utiliser les clés de sécurité, à moins que la sécurité ne soit vraiment nécessaire, et alors se rappeler ce qu'on a assigné.

Exemple :

Les noms de fichiers suivants sont valides :

EXEMPLE
DS1.A
DS0.ABC
FLO.ABCD-X
FLO.ASB-S
FL1.EXEMPLE
FL1.EXEMPLE-S
FL1.EXEMPLE-S:1234

Remarque: Les deux noms de fichiers sont valides, mais ils ne peuvent pas être créés sur le même volume en même temps, parce que les noms sont les mêmes, seules les clés étant différentes.

Les noms de fichiers suivants sont invalides :

- | | |
|--------------|--|
| EXFICHER | - Longueur Maximum du nom dépassé. (7 caractères maximum). |
| FL1EXEMPLE | - Doit être FL1.EXEMPLE. Le point est obligatoire. |
| FLO.EXEMPLE | - L'unité doit être identifiée comme FLO avec le chiffre 0 et non la lettre O. |
| EXEMPLE-AB | - Un seul caractère est utilisé dans la désignation de type. |
| EX-A:12345 | - Quatre caractères seulement sont autorisés dans la désignation de la clé. |
| FL1.A,B-B | - La virgule n'est pas autorisée dans un nom. |
| FL1. EXEMPLE | - Les espaces ne sont pas autorisés dans une ligne de commande. |

1.5.3 Références de Fichiers -- Nom de Programme

C'est un fichier objet qui peut être chargé et exécuté par PROLOGUE. Il a la même syntaxe que le nom de fichier, doit être du type 0 et ne peut pas contenir de référence non résolue. Tous les fichiers du noyau de PROLOGUE entrent dans cette catégorie.

1.5.4 Chargement de PROLOGUE

Lorsque le microordinateur R2E est mis sous tension, le programme initial est automatiquement lu en mémoire et le mot MICRAL.. apparaît sur l'écran. Si cela ne se produit pas, ou si l'on a besoin de recharger le programme, appuyer sur les touches CTRL et SHIFT simultanément pour restaurer l'ordinateur. Bien s'assurer d'avoir un disque ou une disquette PROLOGUE configurée pour le système utilisé.

Pour charger PROLOGUE on procède comme suit :

1. Monter la cartouche de disque ou la disquette sur l'unité qui convient et mettre cette unité en ligne comme indiqué dans le manuel d'utilisateur du système.
2. En réponse à l'appel de MICRAL, frapper :

MICRAL : | L | : [U,] 8 (cr)

Où : B - Charger à partir d'un disque souple
 L - Charger à partir d'un disque dur
 U - numéro d'unité (si omis, le défaut est l'Unité 0)

Remarque : Pour un système sur disque souple seulement, PROLOGUE peut être chargé simplement en frappant (cr) en réponse au message d'appel MICRAL.. Cela suppose qu'une disquette PROLOGUE standard est dans l'unité de disque souple 0.

3. Le système d'exploitation est pris sur l'unité spécifiée à partir de la piste 8 et chargé en mémoire, et le message suivant est affiché sur l'écran :

```
Réalisations Etudes Electroniques
PROLOGUE Version X.X -- JJ/MM/AA
Espace Utilisateur Disponible [XXXX, YYYY]
->
```

Où: X.X - Version de PROLOGUE
 JJ/MM/AA - Date de la configuration sous la forme jour/mois/année.
 XXXX-YYYY - Indique le bloc d'adresses de mémoire disponible aux programmes de l'utilisateur, par exemple, 207E, 7E80 pour un système à 32K. Toute la mémoire de 0000 à XXXX est utilisée par le support de système PROLOGUE. Dans l'exemple donné, 207E est l'adresse de

chargement implicite des programmes d'utilisateur.

- > - Le message d'appel de PROLOGUE indiquant que le système peut maintenant accepter une commande PROLOGUE valide frappée au clavier.

Remarques: Le fichier système PROLOGUE exécutable est un fichier binaire, en général appelé SYST-B. (Il peut, en fait, avoir n'importe quel nom de fichier valide avec le suffixe -B.) Il se trouve toujours sur la piste 8, dans un volume PROLOGUE standard.

Le programmeur système peut placer ce fichier sur une autre piste d'un nouveau volume, si nécessaire. Se référer au Guide du Programmeur Système pour le système d'exploitation, et la Gestion des Fichiers PROLOGUE, publication No R-1004, pour de plus amples détails.

4. Une fois que PROLOGUE est chargé, il reste résident dans la partie basse de la mémoire. Chaque fois qu'on a restauré l'ordinateur (le message d'appel MICRAL... apparaît), il est possible de revenir à PROLOGUE en frappant G/110(cr). Voir Chapitre 7 pour de plus amples détails sur la commande G de chargement automatique.

1.5.5 Syntaxe Générale des Commandes PROLOGUE

En réponse au message d'appel de PROLOGUE, on doit entrer une commande de la forme :

->[&ADR,]NomProgramme[,P1[...Pn]](cr)

Où :

&ADR - Adresse de début de chargement du programme (elle est facultative). Celle-ci est spécifiée par un "et commercial" (&), suivi d'un chiffre hexadécimal, sans préfixe ni suffixe. Exemple : &3000.

Si cette adresse n'est pas spécifiée, le programme est chargé en mémoire à l'adresse implicite de chargement des programmes, contenue dans la position de mémoire 99H (voir paragraphe 7.4 pour plus de détails).

NomProgramme - Doit être le nom d'un programme objet (format défini au paragraphe 1.5.2) qui ne doit pas comporter de référence non résolue. Le programme doit être du type -0, mais la désignation de type n'a pas besoin d'être introduite. Tous les fichiers système PROLOGUE sont dans ce format et figurent dans un répertoire sous la forme Nom-0.

P1...,Pn - Liste des paramètres spécifiés pour le programme. Voir la description de chaque programme PROLOGUE

pour avoir la liste des paramètres applicables.

Cette commande permet à PROLOGUE de charger en mémoire le fichier objet (programme) spécifié, à l'adresse spécifiée (ou à l'adresse implicite de chargement) et de commencer l'exécution du programme.

Il est possible de charger un programme sans exécution immédiate. Voir page suivante pour plus de détails.

Exemples :

Supposons pour ces exemples que l'unité 0 est l'unité de support du système contenant les programmes du système et que l'unité 1 est l'unité de support d'utilisateur contenant les programmes écrits par l'utilisateur.

- >&4000,EXEMPLE(cr) - Un programme appelé EXEMPLE-0 sera recherché sur l'unité 0 (unité système) et chargé en mémoire à partir de l'adresse 4000. Le programme sera ensuite exécuté (si c'est permis par le programme lui-même). Il est à remarquer que PROLOGUE s'attend à trouver le programme exécutable identifié par NomProgramme sur l'unité de support système, sauf indication différente de l'opérateur.
- >CP,SF,TEST-S:2(cr) - L'utilitaire de Copie (appelé en réalité CP-0) sera chargé à partir de l'unité 0 (unité de support système) et exécuté. Il effectuera sa fonction SF de suppression de fichier en cherchant un fichier appelé TEST-S:2 sur l'unité 1 (unité de support utilisateur) pour le supprimer. Comme les unités ne sont pas précisées dans les noms de fichiers, les unités de support par défaut ont été supposées.

Remarques sur la Syntaxe des Commandes

Les règles générales suivantes doivent être observées concernant la liste des paramètres.

1. Tous les éléments d'une liste de paramètres (et d'une ligne de commande) doivent être séparés par des virgules. La ligne de commande contenant la liste de paramètres doit être frappée sans espace entre les éléments.
2. Les paramètres numériques sont en général exprimés en notation décimale. Si on a besoin de nombres hexadécimaux, ils doivent être précédés d'une barre oblique (/), comme dans /F, sauf indication contraire.
3. On peut charger un programme sans exécution immédiate de ce programme de la façon suivante :

- a. Frapper le nom du programme, suivi d'un retour de chariot. Sans Paramètre.
- b. Après que le programme ait été chargé à partir du disque, une virgule (,) apparaît sur l'écran derrière le nom du programme, et le système attend l'introduction des paramètres. A ce moment, on peut changer les volumes sur les unités autant que nécessaire, puis entrer les paramètres requis.

Exemple: Copier une disquette de l'unité de disque souple 0 vers l'unité de disque souple 1.

- a. Charger dans l'unité 0 la disquette système contenant l'utilitaire CP (Copie); une disquette vierge dans l'unité 1.
 - b. Frapper CP (cr).
 - c. Le programme Copie est chargé et l'écran affiche : CP, suivi d'un curseur clignotant.
 - d. Enlever la disquette de l'Unité 0, et monter la disquette à copier.
 - e. Frapper : DV, FLO, FL1 (cr)
 - f. La commande est maintenant exécutée et la disquette recopiée.
4. Un programme du noyau de PROLOGUE qui vient d'être exécuté peut être réexécuté sans nécessité de rechargement. Lorsqu'on frappe le nom du programme, si celui-ci est déjà en mémoire, on se limite à lancer son exécution.

Remarque : les données du programme ne sont pas réinitialisées. Elles gardent l'état qu'elles avaient au sortir de la précédente exécution.

On peut donner cette propriété à un programme écrit par l'utilisateur en langage d'assembleur, en insérant un NOP (code 00) à l'adresse de lancement du programme.

Tous les programmes du noyau de PROLOGUE remplissent cette condition.

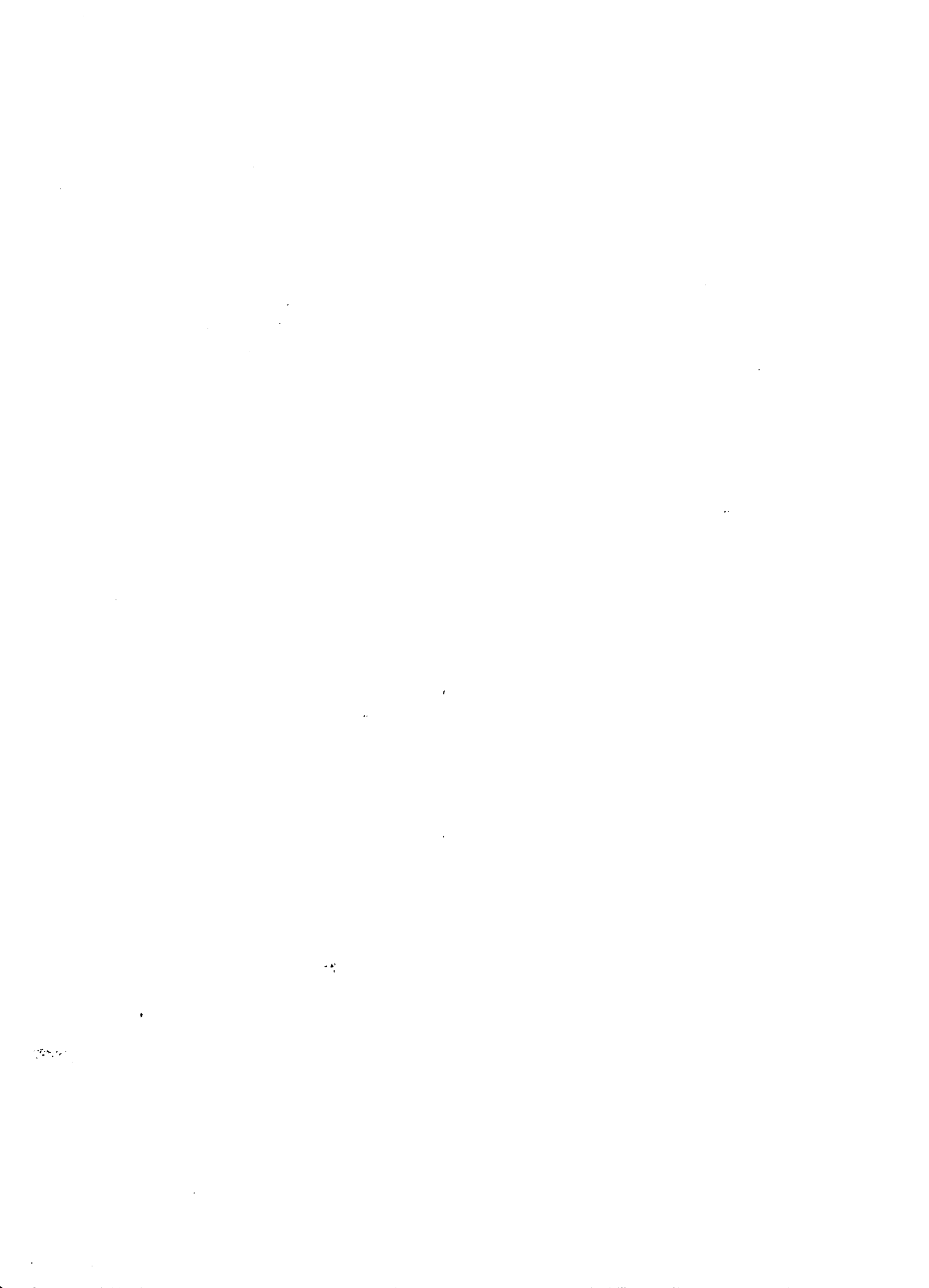
5. En général, le caractère générique "-" indique l'exécution correcte de la commande.
6. Pour sortir d'une commande utilitaire, appuyer sur
 - Touche SORTIE (escape) ou
 - Touche CR (Retour de Chariot ou entrée vide)
7. Pour interrompre une boucle sans fin, appuyer sur le bouton de restauration (ou simultanément sur les touches CTRL-position haute), pour revenir au niveau du chargeur "Bootstrap". Revenir ensuite à PROLOGUE à l'aide de la commande G pour sauter au point d'entrée dans PROLOGUE: MICRAL...G:110(cr).

XOR-423-02568
SRS1008A-998
PROLOGUE 1.9/GBASIC 5.22
Copyright (c) 1983
BULL Micral / MICROSOFT
MICRAL 8022 G - 140K

1.6 DETECTION D'ERREUR

Une fois la commande entrée, l'ordinateur contrôle les erreurs de syntaxe et les erreurs de système (comme les fautes d'orthographe ou la spécification d'une unité non existante). Pendant l'exécution des différentes routines de PROLOGUE, des erreurs (comme les débordements de disque) peuvent être détectées. Lorsqu'une erreur est rencontrée, un code d'erreur est affiché. L'Annexe A donne une liste des codes d'erreur, avec leur définition, et les procédures de recouvrement suggérées.

La détection des erreurs et les codes d'erreur pour chaque processeur de langage sont examinés dans les manuels de référence de langages appropriés.



CHAPITRE 2. COMMANDES ET ROUTINES STANDARDS DE PROLOGUE

2.1 INTRODUCTION

Le Système d'Exploitation sur Disque PROLOGUE comporte un noyau de support standard et des routines utilitaires. Ceux-ci sont fournis avec tous les systèmes PROLOGUE et sont décrits en détail dans le présent chapitre.

D'autres programmes comme le BASIC Microsoft et le BAL peuvent être optionnellement introduits dans le système PROLOGUE. Le fonctionnement de ces programmes sous PROLOGUE est décrit dans les chapitres suivants. L'utilisateur est renvoyé aux manuels de référence particuliers pour les détails sur la programmation.

Tous les programmes supportés par PROLOGUE sont chargés dans l'ordinateur à l'aide d'une ligne de commande qui contient le nom du programme et une série de paramètres. Ceci a été décrit au paragraphe 1.5.5.

A mesure que les programmes sont exécutés, l'ordinateur peut détecter différentes erreurs et afficher un message d'erreurs ou un code d'erreurs. L'Annexe A donne une liste complète des codes d'erreurs avec leur définition et les procédures de recouvrement suggérées.

2.2 PROGRAMMES DU NOYAU DE PROLOGUE

Les programmes PROLOGUE standard décrits dans ce chapitre sont :

<u>Nom Programme</u> <u>Objet</u>	<u>Description</u>	<u>Détails au</u> <u>Paragraphe</u>
/-0	Répertoire -- Affiche le répertoire des fichiers d'un disque ou d'une disquette.	2.3
ED-0	Editeur de Texte--Sert à créer et à modifier des fichiers sources.	2.4
CP-0	Utilitaire de Copie--Sert à créer un volume et à manipuler des fichiers.	2.5
CPS-0	Copie de Secteurs--Sert à copier physiquement tout ou partie d'un disque (créé ou non par PROLOGUE)	2.6

<u>Nom Programme</u> <u>Objet</u>	<u>Description</u>	<u>Détails au</u> <u>Paragraphe</u>
STATUS-0	Statut--Affiche le statut enregistré des unités périphériques magnétiques.	2.7
PATCH-0	Réparation--Utilisé pour examiner et modifier des données enregistrées sur un disque.	2.8
DATE-0	Date--Permet à l'utilisateur d'initialiser la fonction de date et d'heure du système.	2.9
ASG-0	Assignation--Crée un fichier de procédure qui consiste essentiellement en une série de commandes PROLOGUE exécutables pour mettre en oeuvre une série d'opérations. PROLOGUE reçoit ses commandes à partir d'un fichier et non plus du clavier.	2.10
TSM-0	Test des Périphériques Magnétiques--Utilisé pour tester des disques afin de vérifier les opérations.	2.11
TELE-0	Télétransmission--Utilisé pour transmettre des fichiers entre microordinateurs par le canal de communication en série V24.	2.12

2.3 / - REPERTOIRE

Cette commande donne la liste de tous les fichiers contenus dans le volume se trouvant dans l'unité de support spécifiée. Le répertoire peut être sorti soit sur l'écran d'affichage, soit sur une imprimante.

On peut demander la liste de tous les fichiers d'un volume, d'un groupe de fichiers satisfaisant à une référence générique ou un seul fichier.

2.3.1 Répertoire d'un Volume

Cette forme de la commande de répertoire fournit la liste de tous les fichiers d'un volume. Optionnellement, les emplacements (par granule) des fichiers sur le disque peuvent être produits en sortie.

Syntaxe de la Commande :

->[Unité1.]/[,Unité2][,Options](cr)

Où :

- [Unité1.]/ - Spécifie le programme / (répertoire) et, optionnellement, l'unité sur laquelle il se trouve. Le cas par défaut est l'unité de support système.
- [Unité2] - Unité qui contient le volume dont le répertoire doit être listé. Si Unité2 est omis, l'unité de support utilisateur courante est supposée.

Options (dans un ordre quelconque)

- LIS=L0 - Produit la liste sur l'imprimante. L'option par défaut est la liste sur l'écran d'affichage.
- GR - Fournit l'implantation physique sur le disque (par granule) de chaque fichier, suivant la description détaillée ci-dessous.

2.3.1.1 Format d'Affichage

La première ligne d'un affichage de répertoire a la forme suivante :

VOL : Nomvol NOFILES=xx LG.GR : yy OCCUP : n1/n2

Où :

- Nomvol - Nom du volume.
- NOFILES=xx - xx est le nombre maximum de fichiers que ce volume peut contenir.
- LG.GR : yy - Longueur des granules en nombre de secteurs (pour le présent volume).
- OCCUP : n1/n2- n1 = nombre total de granules disponibles sur le volume. n2 = nombre de granules actuellement utilisés. (Evidemment, lorsque ces deux nombres sont égaux, le volume est plein.)

Les fichiers sont listés en plusieurs colonnes et lignes dans l'ordre alphabétique par nom et par type, comme le montre l'exemple ci-dessous. On remarquera que les clés d'accès de sécurité ne sont pas listées dans le répertoire.

Affichage sans l'Option GR

VOL: PR012 NOFILES:63 LG.GR: 16 OCCUP: 40/31

/-0 AZM-0 BASIC-0 COMPRES-C CP-0
CPS-0 ED-0 EDL-0 EXTEND-S MM-0
STATUS-0 SYST-0 TOTO-S

Affichage avec l'Option GR

Si l'on a spécifié l'option GR, le listing du répertoire a le format suivant :

VOL: Nomvol NOFILES=xx LG.GR : yy OCCUP = n1/n2

Nomfichier 1 : A : (B1,C1) [(B2,C2)...]
Nomfichier 2 : A : (B1,C1) [(B2,C2)...]
Nomfichier 3 :

La première ligne de l'affichage est comme déjà vu précédemment. Chaque fichier est listé avec le nom, suivi de la liste des granules assignés aux différents blocs du fichier où:

- A - Taille totale du fichier en nombre de granules
- Bx - Numéro de séquence du premier granule dans le bloc x.
- Cx - Numéro de séquence du dernier granule dans le bloc x.

Ces informations permettent à l'utilisateur de déterminer l'implantation physique précise d'un programme, ce qui est nécessaire dans l'utilisation de plusieurs des routines utilitaires.

Par exemple, supposons qu'un volume soit listé comme suit:

/-0 : 5 : (8,12)
CP-0 : 2 : (16,17)
TR-0 : 4 : (4, 7)
SAM-S : 4 : (14,15) (18,19)

Dans ce cas, /-0 contient 5 granules, les numéros 8, 9, 10, 11 et 12.

Exemple de Commandes de Répertoire

->/ (cr)

- Affichage à l'écran du répertoire de la disquette se trouvant dans l'unité de support utilisateur. Le programme / lu à partir de l'unité de support système.

- >/,FLO,GR(cr) - Liste de tous les fichiers se trouvant dans l'unité de disque souple 0, avec leurs caractéristiques (taille et implantation).
- >/,FLO,LIS=L0,GR(cr) - Comme ci-dessus, mais répertoire listé sur l'imprimante.
- >FL1./,FLO,LIS=L0(cr) - Programme / lu à partir du disque souple 1, répertoire dans disque souple 0 listé sur l'imprimante.

2.3.2 Fichiers Sélectionnés dans Le Répertoire

Cette forme de la commande de répertoire fournit un listing de un ou plusieurs fichiers se trouvant dans un volume.

Syntaxe de la commande :

->[Unité1.]/, [Unité2.]Nomfichier [,Options](cr)

Où :

- [Unité1.] / - Spécifie le programme / (Répertoire) et, optionnellement, l'unité sur laquelle il se trouve. L'option par défaut est l'unité de support système.
- [Unité2] - Spécifie l'unité sur laquelle le volume contenant les fichiers spécifiés doit se trouver. L'option par défaut est l'unité de support utilisateur.
- Nomfichier - Tout nom de fichier valide dans le format standard de PROLOGUE: [Unité.]Nom[-Type][:Clés].

Le nom de fichier peut être spécifique ou générique.

- o Nom de fichier Spécifique -- Nom[-Type], comme dans TEST-0.
- o Nom de fichier Générique -- Au début ou à la fin du Nom, un ou plusieurs caractères et/ou la désignation de type peuvent être remplacés par un astérisque. Dans la recherche et la comparaison des noms, PROLOGUE considère que tous les caractères représentés par l'astérisque sont conformes dans tous les fichiers.

Exemples :

- A*-0 - Spécifie tous les fichiers de Type 0 dont le nom commence par la lettre A.
- *T0-* - Spécifie tous les fichiers de tous les Types dont le nom se termine par les deux caractères T0.
- *-S - Spécifie tous les fichiers de Type S.

Options - Comme décrites dans la commande de répertoire de volume.

Cette commande donne des listings de répertoire pour les fichiers sélectionnés dans les mêmes formats que ceux déjà présentés pour le répertoire de volume en 2.3.1.

Exemples :

- >FL1./,FLO.TEST-*,LIS=L0(cr) - Programme / lu sur disque souple 1, tous les fichiers appelés TEST sur l'unité de disque souple 0 sont listés sur l'imprimante.
- >/,FLO.*-S,GR(cr) - Programme / lu par défaut sur l'unité système, tous les fichiers de type S sur l'unité de disque 0 sont affichés sur l'écran avec la description de leur implantation sur disque (taille, emplacements).

2.4 EDITEUR DE TEXTE

2.4.1 Généralités

L'Editeur de Texte PROLOGUE permet de modifier les fichiers sources existants (types S) et de créer de nouveaux fichiers sources.

Le texte est modifié à partir d'un fichier source et envoyé dans un fichier de destination mis à jour séquentiellement, ligne par ligne, depuis le début du texte jusqu'à la marque de fin de fichier. Chaque ligne peut être modifiée, supprimée ou transférée sans changement. On peut insérer une ou plusieurs lignes isolées ou blocs de texte.

Tous les caractères de texte sont représentés en mémoire dans le code ASCII à 7 bits. Une ligne est terminée par un retour chariot/saut à la ligne, c'est-à-dire par la configuration ODH OAH. Le dernier caractère de tout fichier source doit être la marque de fin de fichier CTRL-Z.

On remarquera que cet éditeur est semblable à l'ancien éditeur BAL utilisé dans les systèmes autonomes, mais n'est pas compatible avec ce dernier.

2.4.2 Syntaxe

Syntaxe de la Commande :

→ Unité.]ED, Nomfichier1[,Options](cr)

où les options sont : { DEST=Nomfichier2 }
CF

A remarquer que les deux fichiers spécifiés doivent être du type S (source). Si la ou les désignation(s) de type sont omise(s), les fichiers sont supposés être des fichiers sources.

Cette commande permet de modifier un programme de deux façons :

1. Si les options sont omises, le programme appelé Nomfichier1 est modifié ligne par ligne et envoyé sur un fichier temporaire (type de nom \$). A tout instant la session de mise à jour, on peut abandonner les corrections, le fichier temporaire est alors abandonné (son nom peut apparaître dans le répertoire) et le fichier original reste inchangé.

Si l'on termine les rectifications normalement et qu'on entre la commande E pour terminer, tout texte restant dans le fichier original est copié sur le fichier temporaire qui est alors rebaptisé Nomfichier1, tandis que l'ancien fichier est supprimé.

2. Si l'option DEST=Nomfichier2 est codée, le programme identifié par Nomfichier1 sert de base pour créer Nomfichier2. A la fin de la mise à jour, Nomfichier2 contient le programme nouvellement mis à jour, tandis que Nomfichier1 reste inchangé et sert à conserver la version précédente.
3. Si l'option CF est spécifiée, Nomfichier1 est créé.

Exemples :

ED,FICHIER,CF	Créer FICHIER-S
ED,FICHIER,DEST=NOUVEAU	Créer NOUVEAU-S à partir de FICHIER-S
ED,FICHIER	Réviser FICHIER-S

2.4.3 Utilisation de l'Editeur

Pour appeler et utiliser l'Editeur :

1. Frapper la ligne de commande appropriée pour charger ED.
2. Quand l'Editeur a été chargé, on verra s'afficher :

RZE EDMIC - Vn.n (Où n.n est le numéro de version)
:_

Les deux-points (suivis par le curseur clignotant) est l'appel de l'éditeur qui demande une ou plusieurs commandes d'éditeur décrites en 2.4.4 ci-dessous.

3. Il faut alors entrer une commande appropriée et commencer à modifier le texte. En général, les commandes de mise à jour sont terminées par un retour chariot. En appuyant sur la barre d'espace, on avance l'affichage à la ligne suivante de la séquence. A mesure que la modification de chaque ligne est terminée (ou qu'elle est sautée sans changement), la ligne suivante est affichée. Il faut remarquer que la mise à jour procède en avançant seulement. On ne peut pas revenir à une ligne précédemment traitée.
4. On notera qu'il est possible de corriger des erreurs de frappe à l'aide des touches suivantes :

<-- (flèche à gauche) - Recule le curseur d'une position sans détruire le caractère sur lequel on passe, permettant ainsi de refrapper le caractère précédent. Peut être utilisé avec la touche REPEAT pour aller rapidement au milieu d'une ligne en vue des corrections.

--> (flèche à droite) - Avance le curseur d'un caractère vers la droite sans détruire le caractère sur lequel on passe.

CTRL-C - Ramène le curseur au début d'une instruction en annulant tous les caractères qui ont été entrés.

5. La touche TAB peut être utilisée dans la modification et l'insertion des commandes. Lorsqu'on appuie sur TAB, le curseur se déplace à droite vers l'arrêt de tabulation suivant. Les arrêts de tabulation sont définis par multiples de 8 caractères sur toute la largeur de l'écran.

Si la machine ne possède pas la touche TAB, cette fonction peut être assignée à l'une des touches de caractères spéciaux par l'un des paramètres du système (voir tableau Chapitre 7). La position de mémoire 8D hexadécimale contient le code de touche qui sera reconnu comme touche TAB. On utilise souvent la barre oblique inverse 5C. (\)

6. Noter que la commande d'éditeur Z spécifie un fin-de-fichier (CTRL-Z) et doit être frappée pour terminer convenablement un nouveau fichier source.

2.4.4 Commandes de l'Editeur

Lorsque l'éditeur affiche ses deux-points d'appel (:), il faut répondre par l'une des commandes d'éditeur décrites ci-dessous. Une commande invalide (par exemple, un X) sera rejetée par le système et n'apparaîtra pas à l'affichage. A la place, un bip se fera entendre.

La toute première commande entrée ne peut pas être M, C, R ou D parce que ces commandes affectent la ligne courante affichée et qu'il n'y en a pas encore pour l'instant.

2.4.4.1 Commandes pour Déplacer Le Curseur

<u>Commande</u>	<u>Fonction</u>
Espace (appuyer sur barre d'espacement) A(cr)	Avancer à la ligne suivante Avance l'affichage à la fin du fichier avec FIN DE FICHIER affiché. La commande I peut alors être utilisée pour insérer du texte supplémentaire, ou les commandes S, E ou Z peuvent être émises pour terminer la mise à jour.

2.4.4.2 Commandes Insertion/Modification/Suppression

<u>Commande</u>	<u>Fonction</u>
	<p>Insérer une ou plusieurs lignes après la ligne courante affichée. Cette commande permet d'ajouter de nouvelles lignes au fichier source. L'Editeur appelle chaque nouvelle ligne par une série de 5 points. En appuyant sur CR, on termine chaque nouvelle ligne entrée et appelle la suivante. En appuyant sur CR après les 5 points, on met fin à la fonction insertion. La ligne qui suit, dans le fichier source, est alors affichée.</p>
R (Remplacement)	<p>Supprimer la ligne actuellement affichée, puis la remplacer par une ou plusieurs lignes entrées exactement de la même façon que par la commande I (insertion).</p>
J : Nomfichier(cr)	<p>Cette commande insère le code source du fichier source (type -S), appelé Nomfichier, immédiatement après l'instruction courante affichée. Le code à insérer <u>doit</u> se terminer par CTRL-Z (commande Z) qui ne sera pas copié.</p> <p>Lorsque le code source a été transféré, les deux-points d'appel sont de nouveau affichés. On passe alors sur le texte, un message FIN D'INSERTION est affiché, et l'on continue à mettre à jour sur le texte original. On notera que les fichiers source et destination doivent tous deux être accessibles pendant cette commande.</p>

<u>Commande</u>	<u>Fonction</u>
M (Modification)	Modifier la dernière ligne affichée. Le curseur est placé au début et l'on peut commencer les modifications. En frappant de nouveaux caractères, on les introduit à la place des caractères affichés à la ligne au-dessus. En appuyant sur la flèche à droite, on déplace le curseur d'une position vers la droite en conservant le caractère sur lequel on passe qui apparaît alors sur la ligne de travail. (La flèche à gauche peut être utilisée pour déplacer le curseur vers la gauche.) En appuyant sur SORTIE (ESC), à tout instant, on déplace le curseur à la fin de l'instruction en conservant les caractères sur lesquels on passe. En frappant un retour chariot, à tout instant, on termine la modification, tous les caractères restant à droite du (cr) étant supprimés. Si nécessaire, M peut être de nouveau entré pour modifier cette ligne.
C (Commentaires)	Similaire à M, mais déplace le curseur à la fin de la ligne de façon qu'on puisse modifier celle-ci ou ajouter des commentaires.

2.4.4.3 Commandes de Recherche de Chaîne

,chaîne(cr)	Permet de trouver l'occurrence de la chaîne dans la partie de texte commençant à partir de la ligne courante.
.chaîne(cr)	
*	,chaîne-- Trouve la première occurrence en n'importe quel endroit du texte de la chaîne spécifiée.
	.chaîne-- Trouve la première occurrence de la chaîne spécifiée, au <u>début</u> d'une ligne.
*	Répète la commande précédente de recherche de chaîne.

L'éditeur recherche dans tout le texte testant dans le fichier (à partir de la présente ligne) jusqu'à ce qu'il trouve la chaîne spécifiée. Toutes les lignes sautées dans la recherche sont transférées sans changement vers le fichier de destination et ne sont pas rafraîchies. La ligne contenant la chaîne spécifiée est affichée lorsqu'elle est trouvée. Si la chaîne n'est pas trouvée, FIN DE FICHER est affiché.

Une chaîne peut avoir jusqu'à 15 CARACTERES, les blancs insérés étant ignorés.

D,chaîne(cr)
D.chaîne(cr)

Similaire aux commandes de recherche de chaîne décrites ci-dessus, sauf que toutes les lignes qui suivent la commande sont supprimées jusqu'à, mais non y compris la ligne qui satisfait la recherche.

D,chaîne - Supprimer les lignes jusqu'à la première occurrence de la chaîne, n'importe où dans le texte.

D.chaîne - Supprimer les lignes jusqu'à la première occurrence de la chaîne, au début d'une ligne.

* - Répéter la commande D-chaîne précédente.

Remarque : Ces commandes peuvent être utilisées pour localiser et extraire des éléments de texte rapidement. Cependant, elles sont à utiliser avec précaution. Une faute de frappe ici peut effacer des parties entières de texte qu'on ne voulait pas supprimer.

2.4.4.4 Fin des commandes de mise à jour

- Z(cr) (fin de fichier) Termine correctement tous les fichiers sources PROLOGUE et doit être émise pour terminer tout programme nouvellement créé.
- Si cette commande est émise pendant la mise à jour d'un programme existant, le fichier de destination sera terminé et tout texte restant dans le fichier source sera ignoré.
- E(cr) (terminer la mise à jour) Cette commande peut être émise à tout instant pendant la mise à jour d'un programme existant pour terminer la fonction de mise à jour. Tout texte restant dans le fichier source sera copié dans le fichier de destination et le système reviendra à l'appel de commande de PROLOGUE Noter que cette commande n'est pas utilisable pour terminer correctement un nouveau programme source.
- S(cr) Abandonne la mise à jour en annulant toutes les modifications au programme source et retourne à PROLOGUE

2.4.4.5 Commande de Longueur de Ligne

- L:nn(cr) Définit une longueur maximum de nn caractères pour les lignes de texte dans les applications comme le FORTRAN où la longueur de ligne est limitée à 72 caractères (image de carte). Si l'on essaie de dépasser cette longueur, le caractère fautif est rejeté et un bip se fait entendre. La longueur par défaut des lignes est de 255 caractères.

2.5 UTILITAIRE CP (COPIE)

L'utilitaire CP est la routine utilisée pour créer un volume PROLOGUE et pour créer et manipuler des fichiers.

2.5.1 Création de Volume

Tous les disques et disquettes utilisés dans le système PRO-LOGUE doivent être formatés (initialisés) au format convenable et recevoir un numéro de volume d'identification. Cette commande peut créer un volume et formater le support ou donner un nouveau nom à un volume existant.

Syntaxe de la Commande :

->[Unité.]CP, CV, Unité, Nomvol [, [LGR=lgr], [NBFIC=nbfichiers], [PM]]

Où :

- | | |
|------------|--|
| [Unité.]CP | - Spécifie le programme CP et optionnellement l'unité sur laquelle il se trouve. La situation par défaut est l'unité de support système. |
| CV | - Indique une création de volume. (Et non une copie de volume). |
| Unité | - Spécifie l'unité sur laquelle le volume doit être créé. |
| Nomvol | - Nom du nouveau volume à créer comprenant jusqu'à 8 caractères. Toute combinaison de lettres, de nombres et de caractères spéciaux est permise, à l'exception de la virgule. |
| Lgr | - Spécifie optionnellement le nombre de secteurs par granule comme étant égal à l'une des valeurs 1-2-4-8-16-32-64-128-0 (en entrant, on spécifie 256). La valeur par défaut est 16. |
| NbFichiers | - Spécifie optionnellement le nombre des fichiers qui peuvent être catalogués dans le répertoire de ce volume. Les premières pistes sur le disque sont réservées au répertoire, avec les besoins d'espace suivants : |
| | Jusqu'à 31 fichiers-1 piste de 16 secteurs |
| | Jusqu'à 95 fichiers-2 pistes |
| | Jusqu'à 159 fichiers-3 pistes |
| | Jusqu'à 191 fichiers-4 pistes |
| | Jusqu'à 255 fichiers-5 pistes |

L'utilisateur ne doit pas déclarer son répertoire plus grand que nécessaire (en particulier sur un disque souple) pour économiser l'espace de travail sur le disque. La valeur par défaut est 63. (Qui exige 2 granules avec la taille par défaut de 16 secteurs par granule--2 pistes.)

PM

- Spécifie optionnellement le formatage (initialisation) de la disquette ou du disque dur. Cela doit être fait avant que le support puisse être utilisé pour la première fois ; en général, il n'est pas nécessaire de le faire à nouveau, à moins que le disque n'ait été effacé.

Lorsqu'on a spécifié PM, le message PREMARQUAGE (O/N) est affiché. On doit confirmer en frappant 0 (oui). Le système formate alors le support (ce qui demande 2 à 3 minutes pour une disquette, 45 minutes pour un disque dur) et revient à l'appel de commande de PROLOGUE lorsque c'est terminé.

Si l'option PM n'est pas spécifiée et que le volume contient déjà des fichiers, le nom du volume existant est affiché sous la forme Nomvolume (O/N):. Si l'on entre 0 (oui), le volume recevra le nouveau nom de volume et les autres caractéristiques spécifiées sur la ligne de commande et les fichiers de l'ancien volume seront supprimés. Cela se produira, même si on donne au volume son nom initial. Pour renommer un volume sans détruire son contenu, on utilisera la commande CP,RV. Toute réponse à la demande de confirmation, autre que 0 provoquera l'abandon de la commande et laissera inchangés à la fois le nom du Volume et les fichiers.

Exemple :

CP,CV,FL1,USER1,LGR=8,NBFIC=159,PM(cr) - L'utilisateur entre la commande et le système lit le programme CP à partir de l'unité de support système.

PREMARQUAGE (O/N) :

- Validation de la demande de formatage. Il faut répondre 0 pour continuer.
- L'utilitaire CP formate le volume sur l'unité spécifiée FL1 en supprimant toute donnée qu'il peut contenir. Le volume reçoit alors le nom de USER1, spécifié pour contenir 8 secteurs par granule, et un espace de répertoire pouvant référencer jusqu'à 159 fichiers.

CP, CV, FL1, USER1, PM(cr)

- Semblable à la commande ci-dessus, mais les valeurs par défaut de 16 secteurs par granule et 63 fichiers au maximum dans le répertoire sont supposés.

2.5.2 Création d'un Fichier

Cette commande crée un fichier sur un volume généré par le système de fichier PROLOGUE. Le nom de fichier est entré dans le répertoire, mais aucun granule ne lui est assigné. Un code d'erreur sera retourné s'il n'y a plus de place dans le répertoire.

Syntaxe de la commande :

-> (Unité.]CP,CF,Nomfichier(cr)

Où :

- [Unité.]CP - Spécifie le programme CP et optionnellement son unité de support.
- CF - Création de fichier.
- Nomfichier - Doit être un nom de fichier (nom, type, clés) qui n'est pas encore employé. Si l'unité n'est pas spécifiée, le fichier est créé dans le volume monté sur l'unité de support utilisateur.

Exemples :

- >CP,CF,EXEMPLE(cr) - Crée le fichier ayant pour nom EXEMPLE, avec le type de fichier et les clés, tous deux en blanc ASCII.
- >CP,CF,EXEMPLE-S(cr) - Crée le fichier ayant pour nom EXEMPLE-S avec les clés en blanc. Le fichier est créé sur l'unité de support utilisateur.
- >CP,CF,FLO.EXEMPLE-S:1 - Crée le fichier sur FLO avec pour nom EXEMPLE-S:1. La clé :1 n'apparaît pas sur l'affichage du répertoire.

Remarque : Un code erreur (voir Annexe A) sera retourné si le nom de fichier spécifié existe déjà sur l'unité spécifiée.

2.5.3 Suppression d'un Fichier

Cette commande supprime le fichier spécifié sur un volume généré par le système de fichier. Les granules assignés à ce fichier sont libérés et le nom de fichier enlevé du répertoire.

Si le fichier existe dans le répertoire avec un type de fichier, ce type doit être spécifié, sinon un code d'erreur "fichier non trouvé" est émis.

Si les clés pour le fichier spécifié sont autre chose que des blancs ASCII (valeur par défaut), elles doivent être spécifiées correctement, sinon un message d'erreur est émis et la suppression n'a pas lieu.

Syntaxe de la Commande :

-> [Unité.]CP,SF,Nomfichier (cr)

2.5.4 Changement de nom d'un fichier

Cette commande a pour effet de changer le nom d'un fichier sans modifier le contenu de ce fichier.

Syntaxe de la Commande :

-> [Unité.]CP,RF,Nomfichier1, Nomfichier2 (cr)

Le fichier identifié par Nomfichier1 sera rebaptisé Nomfichier2. Nomfichier1 doit être indiqué exactement sous la forme où il apparaît dans le répertoire, avec le type de fichier et les clés. Le nom peut être modifié en totalité ou en partie, le type et les clés peuvent être modifiés. L'unité de support spécifiée (s'il y a lieu) doit être la même pour les deux noms de fichier.

2.5.5 Reproduire un Fichier

Cette commande permet de reproduire un fichier de l'une des façons suivantes :

1. Copier d'une unité de support vers une autre, en assignant le même nom ou un nom différent au fichier de destination .
2. Dupliquer un fichier sur la même unité de support en assignant un nom de fichier différent à la copie.
3. Copier d'une unité de support vers l'écran ou une imprimante de ligne.

Syntaxe de la Commande :

-> [Unité.]CP,DF,Nomfichier1 ,Nomfichier2 ou
Unité ou
L0 ou C0

a b

Les données contenues dans Nomfichier1 sont recopiées dans Nomfichier2. Si Nomfichier2 n'existe pas sur l'unité de support spécifiée, il est créé. S'il existe déjà, des données qu'il contenait sont supprimées au préalable. Après la duplication, les deux fichiers ont un contenu identique.

Remarque : il est interdit de dupliquer un fichier sur lui-même.

Les autres options pour le paramètre b sont :

1. Unité - Si seule l'unité de support est spécifiée, le fichier de destination reçoit le même nom que le fichier d'origine.

Il en va de même pour le type et les clés.

Exemple : CP,DF,FLO.FICHER-S,FL1 est équivalent à CP,DF,FLO.FICHER-S,FL1.FICHER-S
CP,DF,FLO.FIC-S:1,FL1.FIC2 est équivalent à CP,DF,FLO.FIC-S:1,FL1-FIC2-S:1

2. CO - Sortie console (unité d'affichage) - option par défaut si Nomfichier2 est omis.

3. LO - Sortie imprimante.

2.5.6 Reproduction d'un Volume

Cette commande a pour effet de reproduire les fichiers contenus dans un volume vers un autre volume.

Syntaxe de la Commande :

-> [Unité.]CP,DV,Vol1,Vol2[,RZ](cr)

Sans Option RZ

L'effet en est la reproduction de tous les fichiers de Vol1 dans Vol2, qui doit avoir été créée au préalable par le système. Tout fichier de Vol2 avec le même nom qu'un fichier de Vol1 est remplacé. Tous les autres fichiers de Vol2 restent inchangés. Avant la copie de chaque fichier, son nom est affiché sur l'écran. Si le répertoire du fichier de destination devient plein pendant cette opération, la commande sera abandonnée et un message d'erreur affiché.

Avec Option RZ

Lorsque l'option RZ est spécifiée dans la commande, tous les fichiers présents sur Vol2 seront supprimés avant que les fichiers soient recopiés à partir de Vol1. Cette opération est validée comme suit :

1. Le message suivant est affiché :

NomVol(O/N) : Où NomVol est le nom du volume de destination.

2. On doit répondre :

- O(cr) (Oui) - Tous les fichiers existant sur Vol2 sont supprimés.
- N(cr) (Non) - La commande est abandonnée sans modification du volume de destination.

Cette opération préserve les caractéristiques du volume de destination :

- * Nom du volume
- * Longueur des granules
- * Nombre maximum de fichiers dans le répertoire.

2.5.7 Changement de nom d'un volume

Cette commande change le nom d'un volume sans changer les caractéristiques physiques de ce volume.

Syntaxe de la Commande :

-> [Unité.]CP,RV,Unité,NouveauNomVol(cr)

Où :

- [Unité.]CP - Programme de copie.
- RV - Spécifie la fonction de Changement de Nom de Volume. (Renommer Volume)
- Unité - Spécifie l'unité sur laquelle le volume visé est monté.
- NouveauNomVol - Nouveau nom à assigner au volume. Peut être toute combinaison contenant jusqu'à 8 lettres, chiffres ou caractères spéciaux (à l'exception de la virgule).

2.5.8 Désignation de groupe de fichiers pour les commandes de suppression et duplication

Pour permettre la recopie facile des groupes de fichiers d'un support sur un autre ou la suppression d'un groupe de fichiers, le programme CP accepte une nouvelle désignation de nom de fichier dans les commandes de duplication de fichiers (DF) ou de suppression de fichiers (SF).

Syntaxe des noms : Un * matérialise un groupe de lettres indéfinies et consécutives qui sont situées soit en tête du nom, soit en fin. Elle peut également matérialiser le suffixe.

Exemple : *A-0 désigne tous les fichiers objets dont le nom se termine par A.

BC*-* désigne tous les fichiers dont le nom commence par BC et de type quelconque.

Exemples d'utilisations dans les commandes

Duplication de fichiers

CP,DF,*31-0,DEST=FLO v duplique sur l'unité de disque souple 0 tous les fichiers objets dont le nom se termine par 31.

CP,DF,AP*--*,DEST=FLO v duplique sur l'unité de disque souple 0 tous les fichiers dont le nom commence par AP quel que soit son type.

Suppression de fichiers

CP,SF,DS1.*-B supprime tous les fichiers Binaires du de l'unité de disque 1.

CP,SF,*-* supprime tous les fichiers du support utilisateur.

2.6 CPS - COPIE DES ERREURS

C'est une routine de copie physique qui copie un disque ou une portion d'un disque sans modification des données. Elle est utile pour la reproduction de disques ou de programmes qui ne sont pas dans le format PROLOGUE, à la différence de l'utilitaire CP qui ne travaille qu'avec des volumes PROLOGUE et avec les notions logiques de fichiers.

Deux types de duplication existent :

1. Mode de copie totale - reproduction d'un support (disque) en totalité, sans modification de la structure, ni de l'emplacement des fichiers. Tout ce qui est enregistré sur le disque de destination est recouvert.
2. Mode de copie libre - reproduction de données sélectionnées par l'utilisateur par piste et secteur ou par adresse de secteur absolue. Cela peut être utile pour copier un petit nombre de secteurs de données à partir d'un disque de sauvegarde, dans un essai de récupération à partir d'une erreur. Ce peut être également utile au programmeur-système pour déplacer des fichiers binaires vers des emplacements différents sur un volume.

2.6.1 Mode de Copie Totale

2.6.1.1. Description

Syntaxe de la Commande :

-> [Unité.]CPS,Unité-S,Unité-D[,PM][,SD][,SDP][,NBPIS]

- Où :
- Unité-S - Unité d'origine, comme par exemple FLO.
 - Unité-D - Unité de destination, comme par exemple FL1.
- PM - Prémarquage (formatage) du support sur l'unité de destination simultanément avec la copie.
- Si l'option PM est omise et que le disque de destination est un volume PROLOGUE, le nom de volume est affiché pour validation de l'opération sous la forme : Nom (O/N). On doit frapper 0 (oui) pour que l'opération de copie se poursuive.
- SD - Spécifie optionnellement une copie à partir d'une disquette simple face vers une disquette double face. Les volumes de destination et d'origine peuvent être les mêmes. Il en résulte une conversion d'une disquette simple face vers une disquette double face.
- SDP - Conversion simple densité de piste -> double densité de piste.
- NBPIS - Convertir seulement le nombre de pistes référencé sur disquette source.

On notera les points suivants :

1. Les disques doivent avoir les mêmes caractéristiques -- nombre de pistes par face ; nombre de secteurs par piste.
2. Le programme CPS préserve les caractéristiques d'une disquette simple face sur la disquette de destination ; c'est-à-dire qu'il ne donne que la capacité de stockage d'une disquette simple face, même sur une disquette de destination double face. Cela peut être corrigé plus tard par la commande RV (Renommer Volume, c'est-à-dire Changer Nom de Volume) de l'utilitaire CP.
3. L'ordre d'entrée des paramètres optionnels n'a pas d'importance.
4. Lorsque la copie commence, le nombre de pistes restant à copier est affiché. Ce nombre est décrémenté régulièrement
5. Lorsque toutes les pistes ont été copiées avec succès, le nombre affiché de pistes restantes sera 0 suivi d'un caractère "-". Un retour aura lieu vers PROLOGUE et la flèche d'appel ---> sera affichée.

6. Comme il existe différents types de lecteurs sur le marché, il convient de référencer ceux utilisés à R2E. Les appellations sont définies comme suit :

- D2 --> lecteur simple face, simple densité de piste
- D4 --> lecteur double face, simple densité de piste
- D8 --> lecteur double face, double densité de piste

Remarque : Ne pas oublier que tous ces lecteurs écrivent et lisent en MFM, codage qui donne double densité d'informations.

De la même façon, un support magnétique (disquette) portera les mêmes références que le lecteur sur lequel elle est générée.

Ex : Disquette D2 : disquette simple face écrite sur un lecteur simple densité de piste.

But : Les lecteurs "D8" ayant une double densité de piste par Inch (96TPI au lieu de 48) il est intéressant de vérifier la compatibilité de relecture d'une disquette D2 ou D4 sur un lecteur D8. Il est entendu que la lecture piste à piste d'une disquette D2 ou D4 correspond à un déplacement de 2 pistes par 2 pistes sur un D8.

La conversion d'un support se fera toujours dans un ordre égal ou croissant du taux d'informations soit :

```

D2 ----> D2 ou
          'D4 ou
          D8
D4 ----> D4 ou
          D8
D8 ----> D8
  
```

Le fichier de conversion est CPS interfacé sous PROLOGUE. Les options à sélectionner sont résumées dans le tableau ci-dessous :

Unité de destination	D2	D4	D8	Disquette Source
D2	PM	/ / / /	/ / / /	avec PM = prémarquage SD = conversion simple face, double face SDP = conversion simple densité de piste, double densité de piste.
D4	PM SD	PM	/ / / /	
D8	PM SD SDP	PM SDP	PM	

Remarque : bien différencier disquette source (support physique) de l'unité de destination (lecteur).

Ce tableau n'est valable que pour un calculateur ayant 2 lecteurs identiques. Il permet suivant les options choisies de lire une disquette écrite par un lecteur autre que ceux présents dans la configuration de la machine et de la recopier.

Une option supplémentaire est disponible dans CPS. Il s'agit de l'option NBPIS.

Cette option est nécessaire pour convertir une disquette dont le nombre de pistes utiles est différent des valeurs standard suivantes :

D2 ---> 35 PISTES
D4 ---> 64 pistes
D8 ---> 160 pistes

Ex : Education Nationale : Supports D2 de 38 pistes

- * Option présente : le nombre de pistes dupliquées sur la disquette de destination sera celui indiqué en référence à la disquette source.
- * Option absente : valeur utilisée = valeur standard déterminée suivant les options SD ou SDP.

Les options sélectionnées peuvent être entrées dans n'importe quel ordre.

Exemple d'utilisation du programme de conversion CPS.

Ex n° 1 Disquette source = D2 --> Unité destination = D8
--> CPS, FLO, FL1, PM, SD, SDP

Ex n° 2 Disquette source = D4 --> Unité destination = D8
--> CPS, FLO, FL1, PM, SDP

Remarque : Si lors d'une conversion l'option de prémarquage est demandée, il reste sur la disquette de destination une zone non écrite. Un prémarquage automatique de cette zone est effectué avec apparition sur l'écran de la visu : "PREMARQUAGE DU RESTE". Dans ce cas, il est possible de transposer ainsi des disquettes sur elle-même.

Ex n° 3 D4 ---> D8 (sur la même disquette)
--> CPS, FLO, FLO, PM, SDP

Les Lecteurs simple face, double densité de pistes, référence : TM 100-3 ne sont pas gérés dans le programme de conversion (double emploi avec les lecteurs D4 étendus à 40 pistes)

2.6.1.2 Affichage d'erreur

En plus des erreurs de système PROLOGUE ordinaires, les erreurs suivantes peuvent être détectées et affichées.

Message d'Erreur

Description

SUPPORTS DIFFERENTS

Les types des unités de support d'origine et de destination ne sont pas identiques (exemple : Disque et Floppy).

SUPPORTS NON-SECTORISES

Tentative de sélectionner une unité système valide qui n'a pas d'arrangement logique en secteur (par exemple, cassette ou bande magnétique).

OPTION INCORRECTE POUR CE SUPPORT

Tentative de recopier une disquette simple face sur une disquette double face, dans un système qui n'a que des unités simple face.

DEBORDEMENT MEMOIRE UTILISATEUR

La mémoire d'utilisateur disponible n'est pas assez grande pour permettre la copie de piste à piste. Essayer de copier un petit nombre de secteurs à la fois, en mode de copie libre.

2.6.2 Mode de Copie Libre

Voir 2.6.2.2 pour la Syntaxe.

2.6.2.1 Description

Ce mode permet de copier une partie d'un disque comme suit :

<u>Origine</u>	<u>Destination</u>	<u>Par</u>
1. Piste quelconque	Piste quelconque	Piste entière spécifiant un nombre de pistes consécutives
2. Piste quelconque et secteur de départ sur cette piste (commençant au secteur si le secteur n'est pas spécifié).	Piste quelconque et secteur de départ sur cette piste (commençant au secteur 0 si le secteur n'est pas spécifié).	Nombre de pistes consécutives ou nombre de secteurs consécutifs.
3. Adresse absolue de secteur quelconque (exemple 0 à 559 pour les disquettes à simple face)	Toute adresse de secteur absolue	Nombre quelconque de secteurs.

La copie s'effectue secteur par secteur, en sorte que le mode de copie libre demanderait un temps très long pour copier un disque complet (560 accès pour une disquette à simple face, par exemple).

2.6.2.2 Procédure

La procédure à suivre pour la copie est indiquée ci-dessous où la première colonne donne les instructions pas à pas et la deuxième montre l'affichage qui apparaîtra sur l'écran. Les réponses de l'utilisateur aux appels de l'ordinateur sont soulignées.

<u>Instructions</u>	<u>Affichage</u>
1. En réponse à l'appel de prologue, frapper CPS(cr).	-> <u>CPS(cr)</u>
2. PROLOGUE charge l'utilitaire CPS, puis renvoie une virgule. Répondre par un retour chariot.	-> CPS(cr), <u>(cr)</u>
3. Le système affiche le mot ORIGINE. Frapper le nom de l'unité d'origine, par exemple FLO, FL1, DS1, etc.	ORIGINE: <u>FLO(cr)</u>
4. Le système demande le numéro de piste d'origine.	
a. En cas de copie par piste et secteur, frapper tout numéro de piste valide.	PISTE: <u>T(cr)</u>
b. En cas de copie par adresse absolue de secteur, frapper (cr).	PISTE : <u>(cr)</u> - L'affichage sera recouvert par le mot SECTEUR :
5. Le système demande le numéro de secteur origine.	
a. En cas de copie par piste et secteur, frapper le numéro du secteur de départ. (Voir note 1).	PISTE : T SECTEUR : <u>S(cr)</u>
b. En cas de copie par secteur absolu, frapper le numéro de secteur de départ. En frappant (cr) au lieu d'un numéro de secteur, on retourne au message d'appel ORIGINE.	SECTEUR : <u>S(cr)</u>
6. Le système demande le nombre de pistes	NB DE PISTES : <u>N(cr)</u>
a. Frapper tout nombre de pistes valide.	
b. Si l'on ne veut pas copier par pistes entières, frapper (cr) Le système demandera le nombre de secteurs. Frapper alors tout nombre valide.	NB DE SECTEURS : <u>S(cr)</u>
7. Le système affiche DESTINATION Frapper le nom de l'unité de destination.	DESTINATION: <u>FL1(cr)</u>

8. Le système demande le numéro de piste de destination (le disque de destination doit être formaté).

a. En cas de copie par piste, frapper tout numéro de piste valide.

b. En cas de copie par adresse de secteur absolue, frapper (cr).

PISTE : N(cr)

PISTE : (cr) - L'affichage sera recouvert par SECTEUR :

9. Le système demande le numéro de secteur de destination.

a. En cas de copie par piste et secteur, frapper le numéro de secteur de départ. (Voir note 1)

b. En cas de copie par secteur absolu, frapper le numéro de secteur de départ. En frappant (cr) au lieu d'un numéro de secteur, on retourne au message d'appel ORIGINE.

PISTE : T SECTEUR : S(cr)

SECTEUR : S(cr)

10. Après que le numéro ait été frappé le système affiche O/N. Frapper 0 pour confirmer la sélection. L'opération a lieu. En frappant N on revient à ORIGINE.

(O/N)
0

11. Lorsque l'opération s'est terminée correctement, un "-" est affiché sur la ligne de commande et le message d'appel ORIGINE est de nouveau affiché.

On peut exécuter une autre copie ou frapper (cr) pour revenir au système d'exploitation .

On notera les points qui suivent :

1. Lorsqu'on copie par piste et secteur, le numéro de secteur spécifié est relatif au secteur 0 de la piste sélectionnée. Ainsi l'on peut spécifier la piste 5, le secteur 100 et la référence aura lieu au centième secteur après le secteur 0 de la piste 5.

2. Les numéros et adresses peuvent être spécifiés en décimal ou en hexadécimal (avec le suffixe H dans ce dernier cas, comme par exemple 1AH).

2.6.2.3 Affichage d'erreur

En plus des erreurs ordinaires du système PROLOGUE, les erreurs suivantes peuvent être détectées et affichées :

<u>Message d'Erreur</u>	<u>Description</u>
NOM D'UNITE INCORRECT	Unité Spécifiée non existante
SUPPORTS NON SECTORISES	Tentative de sélectionner une unité de système valide qui n'a pas un arrangement logique en secteur (par exemple cassette ou bande magnétique).
PARAMETRE INCORRECT	Numéro de piste ou de secteur incorrect (ne peut être que des caractères décimaux ou hexadécimaux valides).
NB EN ENTREE TROP GRAND	Paramètre numérique (ou bien entré directement, ou bien calculé à partir des entrées disque/secteur) dépasse 65.535.

A la suite de l'une de ces erreurs, le message d'appel ORIGINE est retourné et l'on peut essayer de nouveau.

2.7 STATUS

Les données concernant le statut opérationnel de chaque disque sont cumulées dans des compteurs en mémoire, indépendamment du volume. Ces tables sont initialisées à chaque fois que le système est mis sous tension ou restauré, puis enregistre tous les incidents de lecture/écriture qui se produisent dans l'utilisation des unités magnétiques.

Cette routine vérifie et affiche le statut d'un disque donné. Elle permet de vérifier périodiquement l'état opérationnel d'une unité pendant son utilisation. On remarquera que la plupart des erreurs de positionnement tendent à se produire lorsqu'on accède aux pistes les plus élevées sur le disque. Le système d'exploitation réessaiera un accès jusqu'à 10 fois avant d'émettre un code d'erreur. Les échecs sont enregistrés à chaque fois dans les compteurs de statut. Lorsque le nombre des erreurs devient plutôt élevé, il est probablement temps de faire réaligner les unités.

Syntaxe de la Commande :

->[Unité.]STATUS[, [Unité][LIS=L0]](cr)

Où :

Unité - Unité sélectionnée comme FLO: DS1, etc. L'unité par défaut est l'unité de support utilisateur.

LIS=L0 - Sort le statut sur l'imprimante. (Assigner l'imprimante comme unité de listing).

Le statut est affiché selon le format de l'exemple ci-dessous :

SUPPORT FL.1 ERR.xx EN LEC/. NO PISTE P / NO SECT S
EN ECR.

<u>ECHANGES</u>	<u>ERR LEC</u>	<u>ERR ECR.</u>	<u>ERR. POSIT.</u>	<u>SECT. INV.</u>
x	x	x	x	x

Où :

- | | | |
|-------------|---|--|
| SUPPORT | - | Unité vérifiée. |
| ERR xx | - | Indication de la dernière erreur de positionnement lecture/écriture ou xx est le code d'erreur figurant dans la liste de l'Annexe A. |
| NO PISTE P | - | P est la piste où la dernière erreur s'est produite. |
| NO SECT s | - | s est le secteur sur la piste P où l'erreur s'est produite. |
| ECHANGES | - | Nombre total d'échanges lecture/écriture entre la mémoire de cette unité depuis la dernière restauration du système. |
| ERR LEC. | - | Nombre total d'erreurs de lecture. |
| ERR ECR. | - | Nombre total d'erreurs d'écriture. |
| ERR. POSIT. | - | Nombre total d'erreurs de positionnement de piste. |
| SECT. INV. | - | Pour un disque dur seulement, nombre total de secteurs invalides |

Remarques :

1. S'il y a des secteurs invalides pour un disque dur, elles seront listées sous forme de tables par piste et par secteur. Remarquer que le disque dur a deux secteurs de réserve par piste qui sont assignés lorsque des secteurs invalides se produisent sur cette piste. Un maximum de 64 secteurs de réserve seront assignés sur un disque dur.
2. L'erreur 04 sera retournée si le statut de l'unité concernée n'a pas été généré.

2.8. PATCH

2.8.1 Description

Cette routine permet d'examiner et de modifier les données dans un secteur d'un volume sur l'unité de support spécifiée. Cette modification peut être spécifiée de deux façons : 1) par nom de fichier et secteur dans ce fichier ou 2) en spécifiant une piste absolue et un nom de secteur.

Syntaxe de la Commande :

1. Spécification par Nom de fichier :

-> [Unité.]PATCH,Nomfichier[,NSEC=no](cr)

Où :

- Nomfichier - Tout nom de fichier valide. Le type de fichier doit être spécifié.
NSEC=no - Numéro du secteur dans le fichier à modifier où le secteur 0 est le premier secteur du fichier. Un secteur peut être spécifié sous forme décimale ou hexadécimale (avec le suffixe H dans ce dernier cas, par exemple, 10H). (Défaut=0)

2. Spécification par Piste/Secteur :

-> [Unité.]PATCH[,Unité],NPIS=Piste[,NSEC=Secteur](cr)

Où :

- Unité - Unité sélectionnée (Défaut=unité Système)
NPIS=Piste - Numéro piste absolu (Défaut=0)
NSEC=Secteur - Numéro du secteur à l'intérieur de la piste spécifiée (à partir de 0) ; spécifié sous forme décimale ou hexadécimale (avec suffixe H dans ce dernier cas). (Défaut = 0).

2.8.2 Modification du Secteur

Lorsque la commande PATCH est introduite, le système localisera le secteur spécifié et l'affichera sur l'écran sous forme de lignes de 32 caractères hexadécimaux avec la traduction ASCII affichée sur la droite de chaque ligne. Le curseur sera positionné sur le premier caractère du fichier.

On peut alors entrer ce qui suit :

1. Caractères de contrôle du curseur--Se servir des flèches à droite, en haut et en bas pour positionner le curseur sur tout caractère affiché.

2. 0-9, A-F -- Remplacer le caractère hexadécimal sur lequel pointe le curseur par un caractère frappé au clavier. Le nouveau caractère apparaît sur l'écran. On peut continuer à déplacer le curseur et à changer les caractères autant que nécessaire.
3. Une fois les corrections faites, l'une des commandes suivantes peut être entrée (la commande peut être frappée dans n'importe quelle position du curseur, elle ne sera pas affichée) :

- V - Répète l'affichage avec l'équivalent en ASCII des caractères hexadécimaux modifiés. Les modifications ne sont pas encore faites sur le disque.
- I - Imprimer -- Imprime le secteur modifié et son nouvel équivalent ASCII sur l'imprimante et affiche également ces données à l'écran. Les modifications ne sont pas encore faites sur le disque.
- R - Répéter en visualisation -- permet de passer au secteur suivant et de l'afficher à l'écran. Cette procédure s'arrête sur frappe d'un caractère quelconque sur le clavier.
- S - Répéter en impression -- Même procédure que R mais en plus elle imprime le secteur sur l'imprimante.
- M(cr) - Modification - Ecrit le secteur corrigé sur le disque, puis génère la commande V pour afficher le secteur modifié.
- b - (Espace, cr) - Affiche le secteur suivant, en séquence. Si un nom de fichier a été spécifié, cette commande n'affichera pas au-delà du dernier secteur assigné à ce fichier.

SORTIE- Retour au système PROLOGUE.
ESC

Remarquer que PATCH ne doit pas être utilisé dans un fichier binaire absolu. Les changements de valeur auront pour conséquence que le caractère de contrôle CRC généré sera différent du caractère enregistré avec le fichier - en conséquence le fichier ne sera pas chargé. Se référer à la description du moniteur (commande E) dans le Guide du Programmeur Système pour le Système d'Exploitation et la Gestion des Fichiers PROLOGUE si l'on a besoin de modifier un fichier binaire.

2.9 DATE

prologue mémorise la date du jour et entretient l'heure pendant son fonctionnement. Lorsque PROLOGUE est chargé, la date est initialisée comme par exemple 1 janvier 1980, 0 heure, 0 minute. L'utilitaire DATE permet de fixer la date et l'heure courante.

Syntaxe de La Commande :

1. -> [Unité.]DATE, JJ/MM/AA/HH/MM(cr)
- OU
2. -> [Unité.]DATE(cr)
JJ/MM/AA/HH/MM (affichage)

Où : JJ = Jour (1-31)
MM = Mois (1-12)
AA = Année (80-)
HH = Heure (0-23) une horloge de 24 heures est utilisée
MM = Minute (0-59)

Pour l'option 1, on doit entrer le jour, le mois, l'année, l'heure, la minute dans la ligne de commande en deux chiffres pour chaque donnée séparée par des barres obliques et sans espace.

Pour l'option 2, PROLOGUE affiche JJ/MM/AA/HH/MM. On doit alors entrer les chiffres appropriés, suivis par un retour de chariot. Les zéros de gauche ne sont pas nécessaires.

Exemple : -> DATE(cr)
JJ/MM/AA/HH/MM
10/9/80/14/05(cr)

Réponse de PROLOGUE

Lorsque l'entrée est correcte un "-" est affiché à la fin de la ligne de commande et le message d'appel de PROLOGUE est renvoyé. Si l'entrée est incorrecte (jour 30 entré pour Février par exemple), le message INCORRECT est affiché à la fin de la ligne de commande et l'appel de PROLOGUE est renvoyé.

Remarques :

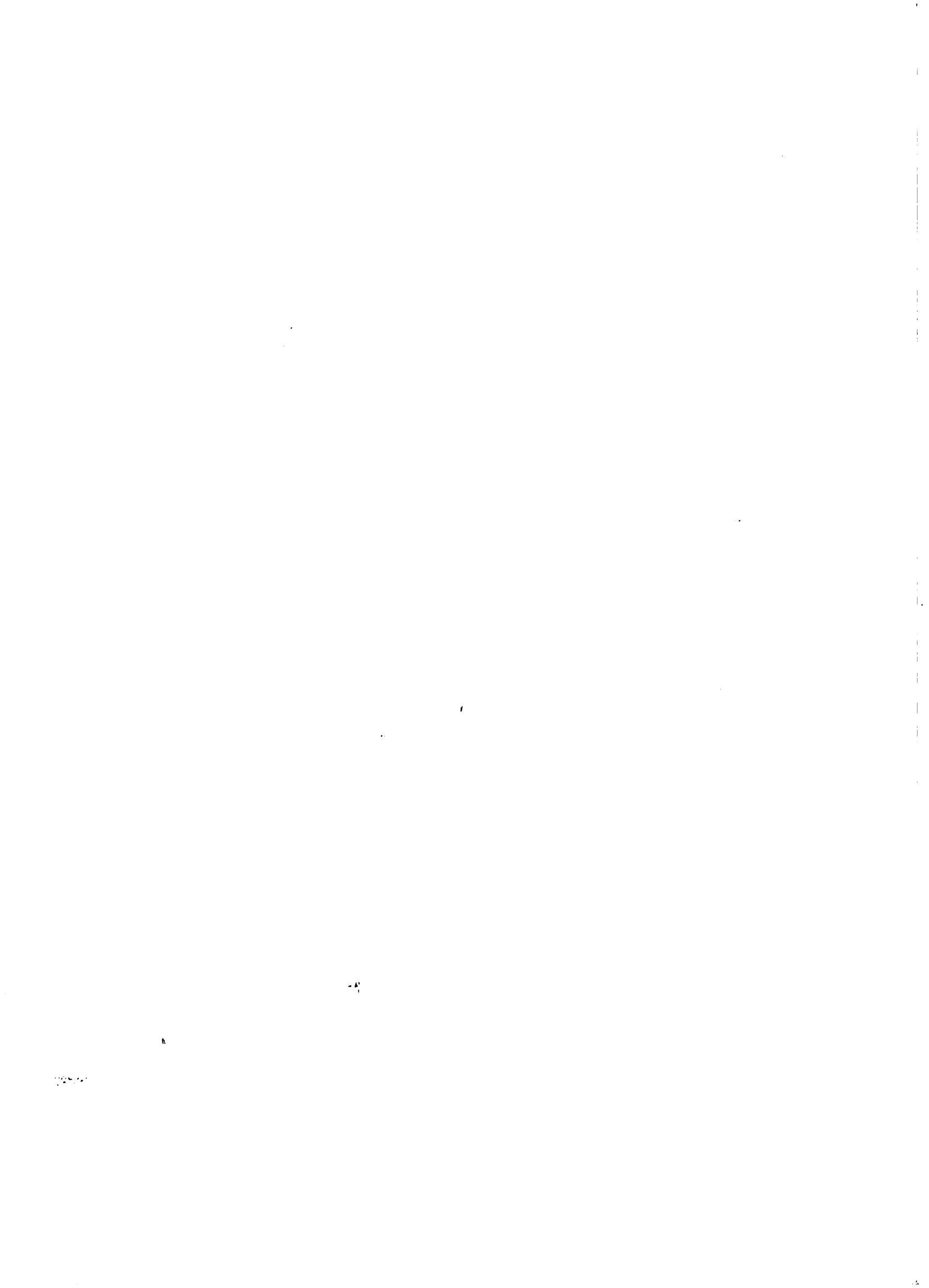
1. Le jour introduit doit être valide pour le mois correspondant. Le 29 février est valide tous les 4 ans.
2. Le programme n'incrémentera pas automatiquement le jour à 23 : 59.
3. Les programmes d'utilisateurs peuvent accéder au compteur de temps et de date par l'instruction PEEK du BASIC. Les données sont stockées en mémoire comme indiqué ci-dessous.

Adresse Hexa

009B
009D
009E
009F
00A1
00A2
00A3
00A4
00A5

Données

ANNEE
MOIS
JOUR
JOUR DE L'ANNEE (0-365)
HEURE
MINUTE
SECONDE
1/10 SECONDE
1/100 SECONDE



CHAPITRE 3. LE BASIC R2E SOUS PROLOGUE

3.1 GENERALITES

BASIC est un langage de haut niveau qui fonctionne sous le contrôle de PROLOGUE. Deux versions sont disponibles :

L'Interpréteur BASIC (nom de fichier BASIC-0) -- Langage interactif dans lequel les entrées de l'opérateur sont soumises à une analyse syntaxique ligne par ligne, à mesure qu'elles sont frappées au clavier. Le programme s'exécute immédiatement après chaque entrée, sans étapes intermédiaires.

Le Compilateur BASIC (nom de fichier BASCOM-0) -- Accepte comme entrée un fichier source écrit au préalable, et le compile pour produire un fichier objet et un listing. Le fichier objet peut alors être exécuté en se servant de l'Editeur de Liens L80.

BASIC contient des facilités pour générer, stocker et manipuler des fichiers de données séquentiels. On peut également utiliser les facilités de fichier direct, de fichier séquentiel et de fichier séquentiel indexé, du Système de Gestion de Fichier, et cela, grâce aux appels à l'Assembleur prévus dans le BASIC.

Le présent chapitre donne des détails sur le chargement du BASIC et l'utilisation des commandes BASIC entrées à partir du clavier. Les instructions et les fonctions du BASIC sont identiques à celles du BASIC Microsoft standard et sont expliquées en détail dans le Manuel de Référence du Programmeur pour le BASIC R2E, numéro de publication R-1005.

3.2 UTILISATION DE L'INTERPRETEUR BASIC SOUS PROLOGUE

Comme mentionné ci-dessus, le programme appelé BASIC est un interpréteur interactif qui analyse vos lignes de programme à mesure qu'elles sont frappées et signale toutes les erreurs rencontrées. Ces erreurs doivent alors être corrigées immédiatement. Un programme peut être entré en mode direct dans lequel les instructions sont exécutées à mesure qu'elles sont entrées (comme dans une calculatrice) ; ou en mode indirect dans lequel le programme est stocké en mémoire et exécuté lorsque toutes les lignes ont été entrées. Un programme entré en mode indirect peut être stocké sur disquette pour une utilisation future.

3.2.1 Invocation de BASIC

Une fois que le système a été lu par appel automatique et que PROLOGUE est chargé, on charge BASIC par la commande PROLOGUE suivante :

```
->[Unité.]BASIC[,Nomprogramme[,F:Nombrefichiers]]
```

Où :

[Unité.]BASIC - Spécifie le processeur de langage BASIC.

- Nomprogramme - Nom de fichier PROLOGUE facultatif, identifiant un programme BASIC précédemment stocké sur le disque. Le nom de fichier doit être dans la forme standard : [Unité.]Nom[-type] [:Clés].
- F:Nombrefichiers - Spécifie le nombre des fichiers qui peuvent être ouverts simultanément. La valeur par défaut est trois ; le maximum est 15.

Lorsque la commande est exécutée, BASIC est chargé et affiche un message de la forme :

```
BASIC REV. 5.11
[BASIC R2E]
Copyright 1977,78,79,80 (C) by Microsoft
Crée JJ-MM-AA
XXXXX Octets Libres
OK
```

XXXXX Octets Libres indique la mémoire disponible pour un programme utilisateur en BASIC.

Le message d'appel OK indique que BASIC est au niveau commande, prêt à accepter des commandes d'utilisateurs. Si le nom de fichier facultatif était spécifié, le programme en BASIC ainsi identifié sera en mémoire, prêt à s'exécuter.

Exemples :

- >BASIC(cr) - charge l'interpréteur BASIC ; spécifie implicitement que 3 fichiers peuvent être ouverts simultanément.
- >BASIC,FICHER-S - Comme ci-dessus, mais charge et exécute le programme FICHER-S.
- >BASIC,FICHER-S,F:5 - Comme ci-dessus, mais déclare un maximum de 5 fichiers ouverts simultanément.

3.2.2 Modes d'Opérations

BASIC peut être utilisé en deux modes :

- Direct - Lorsque les instructions et les commandes sont entrées sans être précédées de numéros de lignes, elles sont exécutées immédiatement et OK est retourné après une exécution correcte. Les résultats des opérations arithmétiques et logiques peuvent être affichés immédiatement et stockés pour une utilisation ultérieure, mais les instructions elles-mêmes sont perdues après l'exécution. Ce mode est utile pour la mise au point et pour les calculs rapides qui ne demandent pas un programme complet.
- Indirect - Ce mode permet d'entrer des programmes. Chaque ligne de programme est précédée par un numéro de ligne et

se trouve stockée en mémoire. On exécute le programme stocké en entrant la commande RUN. Le programme peut également être stocké sur disque par les commandes SAVE et RUN. Voir paragraphe 3.2.4 pour une description de ces commandes.

3.2.3 Numéros de Lignes

Les numéros de lignes indiquent l'ordre dans lequel les lignes sont stockées et exécutées, et servent aussi comme référence dans les branchements et les mises à jour. Les numéros de ligne doivent être compris entre 0 et 65529. Les lignes n'ont pas besoin d'être entrées dans l'ordre numérique croissant, mais elles apparaîtront dans le programme dans l'ordre numérique croissant correct. On peut modifier une ligne en réentrant son numéro et en frappant les nouvelles informations, ou bien en utilisant la commande EDIT de BASIC. (Voir Manuel de Référence BASIC pour plus de détail.)

C'est une bonne habitude de numéroter les lignes par 10, en laissant ainsi de la place pour insérer des numéros de lignes.

3.2.4 Commandes de Programme/Fichier BASIC

BASIC peut fonctionner à un niveau commande et à un niveau programme. Au niveau commande, BASIC s'attend à recevoir des commandes venant du clavier. La structure de ces commandes est décrite ci-dessous. Comme elles sont compatibles avec le système d'exploitation PROLOGUE, les commandes spécifiant des fichiers diffèrent légèrement, au point de vue de la syntaxe, du BASIC Microsoft standard.

Les noms fichiers utilisés doivent être dans le format PROLOGUE standard :

[Unité.]Nom[-Type][:Clé].

Si le type n'est pas spécifié, le type -S (source) est supposé.

Les instructions BASIC utilisées au niveau programme sont identiques au BASIC Microsoft standard. Voir la publication R-1005, Manuel de Référence BASIC, pour plus de détails.

3.2.4.1 Commande RUN

Syntaxe de la Commande, Forme 1 :

RUN [numéro de ligne]

Cette commande est frappée après l'entrée au clavier d'un programme en BASIC. L'exécution de ce programme commence à partir du numéro de ligne le plus petit ou du numéro de ligne facultatif spécifié dans la commande RUN. BASIC retourne au niveau commande après l'exécution d'un RUN.

Syntaxe de la Commande, Forme 2 :

RUN "Nomfichier"[,R]

Charge le fichier spécifié à partir du disque et l'exécute. RUN

ferme tous les fichiers et supprime le contenu courant de la mémoire avant de charger le fichier désigné. Cependant, si l'option R est spécifiée, tous les fichiers de données restent ouverts.

Exemple : RUN"FLO.NEW",R -- Charge NEW à partir du disque souple 0 après avoir supprimé le contenu courant de la mémoire, laisse tous les fichiers de données ouverts, et exécute le programme NEW.

3.2.4.2 Commande SAVE

Syntaxe de la Commande :

SAVE"Nomfichier"[,A] ou SAVE "Nomfichier"[,P]

Cette commande écrit le programme BASIC actuellement en mémoire, dans le fichier spécifié sur disque. Si l'unité n'est pas spécifiée dans le nom du fichier, l'unité de support utilisateur est supposée. Si le type n'est pas spécifié, le type -S (source) est supposé. Si un fichier avec le nom de fichier spécifié existe déjà, il est recouvert par l'écriture nouvelle.

Lorsque l'option A est spécifiée, le fichier est sauvegardé en format ASCII et peut être mis à jour par la routine de modification (ED) de PROLOGUE. Si A n'est pas spécifié, le fichier est sauvegardé dans la forme binaire condensée qui exige moins d'espace disque, mais qui est inintelligible pour l'Editeur de PROLOGUE. On remarquera que si un fichier doit être utilisé dans une commande MERGE de BASIC, il doit être en format ASCII.

Remarquer que la commande LOAD est capable de traiter l'un ou l'autre type de fichier.

On utilise l'option P pour protéger le fichier en le sauvegardant dans un format binaire codé. Lorsque, par la suite, un fichier protégé est exécuté (RUN) ou chargé (LOAD), toute tentative pour le lister ou le mettre à jour sera rejeté par BASIC.

Exemple : SAVE"FL1.NEW-S",A -- Sauvegarde le programme BASIC en mémoire en l'écrivant en format ASCII sur la disquette dans l'unité de disque souple 1 comme fichier source appelé NEW-S.

3.2.4.3 Commande LOAD

Syntaxe de la Commande :

LOAD"Nomfichier"[,R]

Cette commande charge en mémoire le fichier spécifié à partir du disque. Si le type n'est pas spécifié, un fichier de type -S est chargé par le programme.

LOAD ferme tous les fichiers ouverts et supprime le programme actuellement en mémoire avant de charger le programme désigné. Une

fois le programme chargé, BASIC retourne au niveau commande. Si l'option R est spécifiée, le programme est exécuté (RUN) après son chargement, et tous les fichiers ouverts restent ouverts.

3.2.4.4 Commande MERGE

Syntaxe de la Commande :

MERGE "Nomfichier"

Cette instruction fusionne le fichier disque spécifié dans le programme actuellement en mémoire. Aux endroits où le nouveau programme contient des lignes avec le même numéro de ligne que les lignes déjà en mémoire, ces dernières sont remplacées.

Le fichier spécifié doit avoir été sauvegardé (par SAVE) en format ASCII, sinon une erreur "Mauvais Mode de Fichier" se produit. Si le type de fichier n'est pas spécifié, le type -S est supposé.

3.2.4.5 Commande KILL

Syntaxe de la Commande :

KILL "Nomfichier"

KILL supprime du disque le fichier spécifié. Si une tentative est faite pour supprimer un fichier ouvert, une erreur "Fichier Déjà Ouvert" se produit. KILL peut supprimer tout fichier, créé sous BASIC ou non.

3.2.4.6 Commande NAME

Syntaxe de la Commande :

NAME "anciennomdefichier" AS "nouveaunomdefichier"

Cette commande donne au Fichier de nom : anciennomdefichier le nom : nouveaunomdefichier.

Le fichier désigné par anciennomdefichier doit exister.

Le nouveau nom ne doit pas être déjà présent sur le support et doit suivre la syntaxe habituelle des noms de fichiers.

Cette instruction modifie uniquement le nom du fichier sans toucher à son contenu.

3.2.4.6 Commande FILES

Syntaxe de la Commande :

FILES "unité."

La commande agit comme suit :

FILES (cr)	Affiche le catalogue de l'unité utilisateur
FILES "unité."	Affiche le catalogue de l'unité désignée.

Exemples :

FILES "FL1." - Affiche les noms des fichiers présents sur l'unité de disque souple n°1.

3.2.4.8 Autres Commandes

Les commandes NEW, LIST et SYSTEM sont codées exactement comme dans le BASIC Microsoft.

NEW supprime le programme actuellement en mémoire et remet à zéro toutes les variables.

LIST[n°ligne]-[n°ligne]-- Cette instruction liste le programme actuellement en mémoire à partir du numéro de ligne de départ facultatif (ou du numéro de ligne le plus petit) au numéro facultatif de ligne de fin (ou au numéro de ligne le plus élevé).

LLIST -- Similaire à LIST, mais la liste est produite sur l'imprimante

SYSTEM -- En frappant cette commande, on provoque un retour à PROLOGUE.

3.2.5 Instructions Mettant en Jeu des Fichiers de Données

Plusieurs instructions BASIC concernent des manipulations de fichiers de données. Ce sont OPEN, PRINT#, PRINT# USING, INPUT, LINE INPUT#, WRITE#, CLOSE, FIELD, LSET/RSET, EOF, LOC. Lorsque ces instructions spécifient des noms de fichier, ces noms de fichier doivent être dans le format PROLOGUE standard : [Unité.]Nom[-Type] [:Clés]. Se référer au Manuel de Référence BASIC, publication numéro M-1005 pour une description complète de ces instructions.

3.3 UTILISATION DU COMPILATEUR BASIC MICROSOFT SOUS PROLOGUE

Syntaxe de la Commande :

```
->BASCOM Fichiersource, [ Fichiertrans ], Fichierliste [ ,options ]
          |          |          |          |          |
          CI         'NR         LO         CO         NL
```

Où :

BASCOM - Compilateur BASIC

Fichiersource - Tout nom de fichier source valide en ASCII. Exemple : MONPROG. (Le type par défaut est -S).

CI - Source entrée directement au clavier.

- Fichiertrans - Nom du fichier translatable à générer ; le type est -R (relocatable, c'est-à-dire translatable). En cas de défaut, ce fichier reçoit le même nom que le fichier source, avec le type -R. Exemple : MONPROG-R.
- NR - Indique qu'il n'y a pas de génération de fichier translatable.
- Fichierliste - Nom du fichier de sortie du listing. Le type est -L.
- CO - Listé sur écran.
- LO - Listé sur imprimante.
- NL - Pas de listing.
- Options
- E - Traiter les instructions ON ERROR GOTO.
- X - Traiter les instructions RESUME.
- N - Sans listing du code machine généré
- D - Traiter les instructions de mise au point (TRON/TROFF)

Remarque: Le Compilateur BASIC MICROSOFT produit des fichiers translatables dans un format Microsoft. Ces fichiers doivent être édités par l'Editeur de Liens Microsoft L80 pour produire un code translatable au format R2E (fichier objet de type 0)

3.4 UTILISATION DE L'EDITEUR DE LIEN MICROSOFT L80 SOUS PROLOGUE

Syntaxe de la Commande :

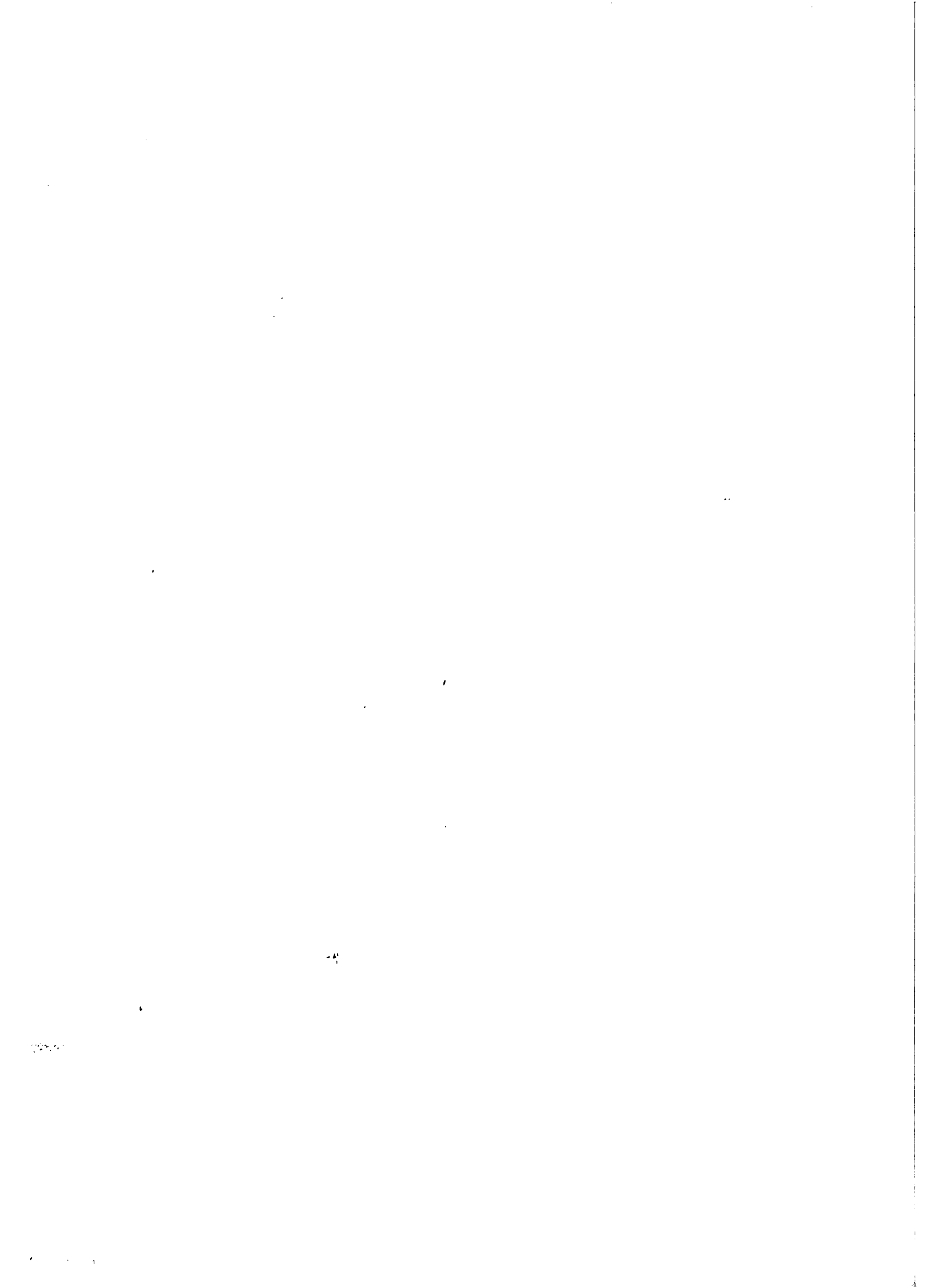
```
->L80(cr)
  *Fichiertrans,...,Bibliothèque/S,Fichierobjet/N/E
```

Où :

- * - Caractère d'appel * indique la fin du chargement
- Fichiertrans - Fichier(s) translatable(s) produit(s) par BASCOM. Le type par défaut est -R.
- Bibliothèque/S - La bibliothèque pour BASCOM est BASLIB-R.
- Fichierobjet - Nom du fichier objet à produire. Le type par défaut est -0.

Exemple :

```
---> BASCOM,MONPROG           entrée:MONPROG-S,Sortie:MONPROG-R
---> L80
*   MONPROG,BASLIB/S,MONPROG/N/E  entrée:MONPROG-R,Sortie:MONPROG-O
```



CHAPITRE 4. BAL SOUS PROLOGUE

4.1 GENERALITES

BAL est le Langage BASIC Orienté Gestion de R2E -- c'est un sur-ensemble puissant du BASIC ANSI. Il offre des instructions renforcées pour l'entrée/sortie des informations orientées gestion et contient des instructions pour le Système de Gestion de Fichiers optionnel (gestion de fichiers directs, séquentiels et séquentiels indexés).

BAL est un langage de compilateur qui demande un fichier source BAL comme entrée dans un module traducteur. Le traducteur (appelé TR) traduit ce fichier source et produit un fichier intermédiaire. Ce fichier ne peut être chargé et exécuté que sous le contrôle du module exécuteur BAL (appelé EX).

On peut donner à ce fichier source tout nom de fichier valide et l'écrire dans le format BAL en se servant de l'Editeur PROLOGUE (ED) décrit au paragraphe 2.4. Le fichier source reçoit le type implicite -S si un autre type n'est pas spécifié.

Se référer à la publication numéro R-1006, Manuel de Référence BAL, pour des détails complets sur l'ensemble des instructions BAL et le format requis pour un programme BAL.

REMARQUE: Une version autonome de BAL existe. Cette version n'opère pas sous le contrôle de PROLOGUE, et les programmes autonomes BAL ne sont pas compatibles avec BAL sous PROLOGUE.

4.2 LE TRADUCTEUR BAL - TR

Lorsqu'un programme BAL a été préparé dans le format convenable à l'aide de l'éditeur PROLOGUE, il doit être traduit par la routine de traduction BAL qui est cataloguée sur un disque BAL sous le nom de TR-0. La commande de traduction s'utilise comme suit :

Syntaxe de la Commande :

->[Unité.]TR,Nomfichier[,Options]

Où :

- [Unité.]TR - Spécifie le programme de Traduction.
- Nomfichier - Identifie un fichier source BAL. Le nom du fichier doit être dans le format PROLOGUE correct : [Unité.]Nom[-Type][:Clés]. Si Type est omis, le type -S est supposé.
- Options - Plusieurs options de compilation peuvent être spécifiées dans n'importe quel ordre, séparées par des virgules. Ce sont :

<u>Option</u>	<u>Définition</u>	<u>Cas par Défaut</u>
NL	Sans Listing	Listing sur écran si
LIS=LO	Listing sur l'imprimante	aucune option de lis-

	de ligne	te n'est spécifiée
ND	Sans Mise au Point -- Les adresses de mise au point (Debug adress) ne sont pas sorties sur le listing du programme	Adresses de mise au point fournies
TP	Traduction partielle	Traduction complète
DEST=Nomfichier	Spécifie le nom du fichier intermédiaire obtenu	Le fichier intermédiaire reçoit le même nom que le fichier source, avec le type de fichier -T

Cette commande a pour effet de traduire le fichier source spécifié avec les options indiquées. On notera les points suivants :

1. Si ND est omis, le listing du programme est produit avec les adresses de mise au point (debug address), calculées et listées à droite de chaque ligne de programme. Ce sont les adresses effectives en mémoire du code de cette instruction. Ces adresses sont nécessaires lorsqu'on utilise l'option de mise au point (Debug option) pour la correction du programme. Voir le Manuel de Référence BAL, R-1006, pour des détails complets.
2. Si TP, traduction partielle, est spécifié, on peut donner seulement certains des segments du programme à traduire. Il faut alors fournir ces numéros de segment comme suit :

Numéro du segment :

Cette option est utile si plusieurs segments du programme travaillent correctement et qu'on ne veut traduire qu'un ou plusieurs segments dans lesquels les erreurs ont été corrigées.

3. En spécifiant DEST=Nomfichier, on peut assigner tout nom de fichier valide au fichier BAL intermédiaire.

Lorsque des erreurs de programme sont détectées pendant la traduction, elles sont traitées comme suit :

1. Si le listing est sur l'écran, un message d'erreur est affiché et la sortie s'arrête, donnant à l'utilisateur le temps de noter l'erreur. Le caractère fautif est mis entre parenthèses, et un numéro d'erreur (voir Manuel de Référence BAL) est donné sur la liste.

En appuyant sur SORTIE (ESC), on continue la traduction.

2. Si le listing est sur l'imprimante, un message d'erreur est imprimé sous l'instruction en erreur, et la sortie continue.

Par exemple : DCL A,B,CD
 ***(D) ERROR 61 DEBUG ADRESS 0007

Lorsque l'erreur rencontrée pendant la traduction est fatale, on voit une longueur de programme de zéro listée à la fin de la traduction.

Exemples :

1. TR,FLO.NEW,LIS=LO Programme traducteur chargé à partir de l'unité de support Système ; fichier source BAL, NEW-S (S supposé par défaut) chargé à partir de l'unité de disque souple 0 et traduit ; fichier intermédiaire généré sur unité de support utilisateur et appelé NEW-T par défaut ; listing sorti sur imprimante.

2. FL1.TR,FLO.NEW,ND,
 DEST=NEW2-T Traducteur chargé à partir de l'unité de disque souple 1, fichier source NEW-S chargé à partir de l'unité de disque souple 0, fichier intermédiaire appelé NEW2-T généré sur unité de support utilisateur, listing sorti sur écran d'affichage par défaut.

Remarque :

Pour arrêter la traduction, appuyer sur la touche SORTIE (ESC).
Pour revenir à PROLOGUE (après SORTIE), appuyer sur la touche R.

--> CV/SBAL : conversion de programme Source BAL de système
 BAL à PROLOGUE

 syntaxe : --> CV/SBAL,nomsupport,NPIS=n \times piste,nomfichier v

Nom support : support origine créé sous système BAL

NPIS : n \times piste piste d'implantation du Source à transférer

En général : disquette D2 (140K) : piste 0
 : disquette D4 (256K) : pistes 17,23,29,35
 : disque D120 : pistes 14,16,.....,100,102

longueur maximum d'une ligne source : 255 octets.

exemple :

--->CV/SBAL,FL1,0,FLO.PGM1-S : Convertir Source de
 disquette FL1,piste 0 en
 fichier PGM1

--->CV/SBAL,DS1,16,D50.PGM2-S : Convertir Source BAL de
 disque unité 1 piste 16 en
 fichier PGM2 sur unité 0

Note : Dans la conversion des programmes sources écrits en BAL et utilisant les fichiers, les utilisateurs feront attention au changement des codes-retours listés ci-après.

--> CV/FBAL Conversion d'un fichier du Système BAL à PROLOGUE

Syntaxe : --> CV/FBAL, support, fich-orig, LGR=nb secteurs, fic-dent [options] v

où support : type support origine ex : FLO, FL1, DS0, DS1
créé sous système BAL

fichier origine : fichier créé sous système BAL
(nécessairement de type séquentiel ou relatif, les fichiers séquentiels indexés ne sont pas convertibles directement).

LGR : longueur des granules en nombre de secteurs, sur le support créé sous BAL

disquette LGR = 16

D120 LGR = 256

Option

NSEC : n^e secteur adresse du début du système de gestion de fichier sur support BAL

défaut NSEC = 0, (floppy)

NSEC = 2180H (notations hexadécimales système 8030, 8031, 8040)

NSEC = 2500H (système multiposte 8050K, 8060)

Exemple : CV/FBAL, DS1, CLIENTS, LGR = 256, DS0.CLIENTS, NSEC=2180H

Conversion fichier CLIENTS dans système BAL
monoposte de disque 1 à disque 0 sous fichier PROLOGUE.

Correspondance de codes PROLOGUE /BAL.SGF

En décimal

PROLOGUE	BAL	Signification
40	70	fichier inexistant
41	82	fichier déjà existant
43	72	fichier déjà ouvert
46	85	débordement de volume
47	71	clés d'accès incorrects
48	100	fin de fichier
50	73	numéro logique incorrect
53	84	adresse disque inconnue
54	68	volume non ciré par le SGF
56	64	module non présent dans le système
60	96	type du fichier incorrect
62	101	zone tampon de longueur différente des enregistrements
64	99	début de fichier
65	115	tentative de suppression/écriture de fichier sur bande magnétique autre que le dernier
70	102	longueur de clé trop petite
71	102	" " trop grande
73	105	index nul (interdit)
74	113	clé bloqué par un autre utilisateur
75	119	débordement du fichier clé
76	86	fichier clé incohérent
77	117	débordement du fichier de données
78	104	clé non trouvée
79	103	clé trouvée mais index faux
81	112	clé déjà inexistante.

4.3 L'EXECUTEUR BAL - EX

L'Exécuteur BAL (catalogué sur le disque sous le nom de EX-0), est un progiciel qui conduit l'exécution des programmes BAL traduits. La commande Exécute s'utilise comme suit.

Syntaxe de la Commande :

->[Unité.]EX, NomfichierInt[, D]

Où :

- [Unité.]EX - Spécifie le progiciel Exécuteur de BAL.
- NomfichierInt - Identifie le fichier BAL intermédiaire produit par le programme TR. Peut être n'im-
porte quel nom de fichier valide. Le type
-T est supposé comme cas par défaut.
- [D] - Spécifie une exécution du programme sous le
contrôle de DEBUG. Voir le Manuel de Réfé-
rence BAL pour une description complète de
l'utilisation du progiciel DEBUG.

Lorsque cette commande est exécutée, l'Exécuteur BAL est chargé en mémoire, puis le programme BAL spécifié est chargé en mémoire et exécuté. Si une erreur est rencontrée pendant l'exécution, un message d'erreur est affiché (voir Manuel de Référence BAL pour la liste des messages d'erreur), le programme BAL est abandonné et l'on retourne au message d'appel de PROLOGUE.

Un message d'erreur aura la forme :

ERREUR n DANS SEGMENT x A L'ADRESSE yy

- Où : n est le numéro de l'erreur.
x est le segment de programme.
yy est l'adresse de mise au point à l'intérieur du segment.

CHAPITRE 5. FORTRAN-80 MICROSOFT SOUS PROLOGUE

Syntaxe de la commande :

>F80, {fic.Source} [, {fic-résultat}] [, {fich.listage}] [, Options] v

{fic.Source} :
 CI

{fic-résultat} :
 NR

{fich.listage} :
 LO
 CO
 NL

Options :

où :

Fich-Source : fichier Source de type - S par défaut

CI : source entré directement au clavier

fichier-résultat : résultat de la compilation type rélogeable Microsoft par défaut -R

NR : pas de génération de fich-résultat

fich-listage : fichier format listage/imprimante

LO : imprimante

CO : écran visuel

NL : pas de fichier-listage

options

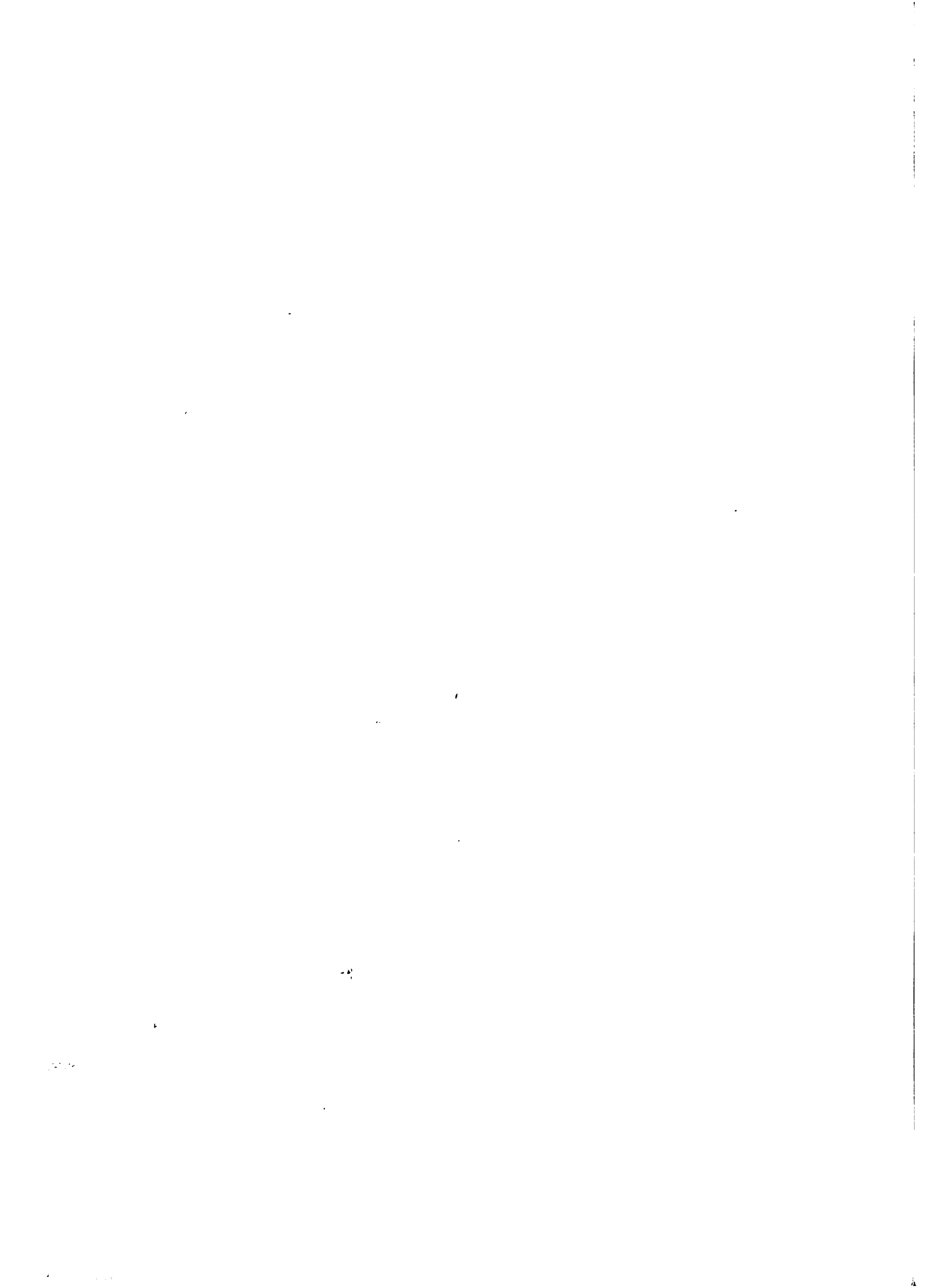
N : pas de génération des instructions assembleurs sur le fichier-listage.

Exemple :

--> F80,PI v traduit le programme Source PI-S en fichier rélogeable Microsoft PI-R

--> L80 v

* PI,FORLIB/S,PI/N/E édition de lieu Microsoft produisant le programme executable PI-O à partir du fichier PI-R et de la bibliothèque FORLIB-R.



CHAPITRE 6. COBOL-80 MICROSOFT SOUS PROLOGUE

Syntaxe d'appel :

```
-->COBOL, {fic.Source} [ , [ fic-résultat NR ] , [ fich.listage LO CO NL ] ] v
```

où :

Fich-Source	:	fichier Source de type - S créé pour ED ou WM
CI	:	source entré directement au clavier
fichier-résultat	:	résultat de la compilation en format rélogeable Microsoft de type -R
NR	:	pas de génération de fichier -R
fich-listage	:	fichier format listage/imprimante
LO	:	Sortie imprimante
CO	:	Sortie écran
NL	:	pas de sortie

Exemple :

```
--> COBOL, CARRE v          Compile CARRE-S en CARRE-R
```

```
--> COBOL v                (Utiliser la version de L80 permettant la segmentation)
```

* CARRE, COBLIB/S, CARRE/N/E S'assurer de la présence du fichier

Contrôleur écran : CRTDRV-R sur le disque utilisateur

CHAPITRE 7. ORGANISATION DU SYSTEME DE FICHIERS

7.1 GENERALITES

Ce chapitre donne à l'utilisateur les informations générales concernant les caractéristiques de PROLOGUE. Une section particulièrement utile est la section 7.4 qui décrit les différents paramètres du système. Par exemple, en examinant le contenu des positions de mémoire 99H et 100H, on peut déterminer l'adresse de chargement de tout programme chargé par PROLOGUE. Ainsi, supposons qu'une erreur se soit produite et qu'il faille redémarrer le programme ; pour redémarrer le programme à l'emplacement spécifié, on restaure le système et on répond au message d'appel MICRAL en frappant G:XXX, où XXX est l'adresse de départ. (L'adresse de départ de PROLOGUE est 110).

Pour des détails complets sur l'interface avec PROLOGUE comprenant les informations sur le format des fichiers de données et les routines en assembleur constituant PROLOGUE, se référer au Guide du Programmeur pour le Système d'Exploitation et Le Système de Gestion des Fichiers de PROLOGUE, numéro de publication R-1004.

7.2 NOYAU DU SYSTEME DE GESTION DES FICHIERS

Le noyau ou coeur du système de gestion des fichiers est constitué par un ensemble de fonctions élémentaires qui gèrent les fichiers sur les unités de support système. Celles-ci créent le début d'une structure dynamique de fichiers et contrôle l'allocation d'espace disponible sur le disque ou la disquette, et la lecture et/ou l'écriture des fichiers.

7.3 ORGANISATION DE L'ESPACE DISQUE

Chaque disque et disquette est formé d'une série de pistes concentriques, chacune desquelles est divisée en secteurs de 256 caractères, comme le montre la figure 7-1.

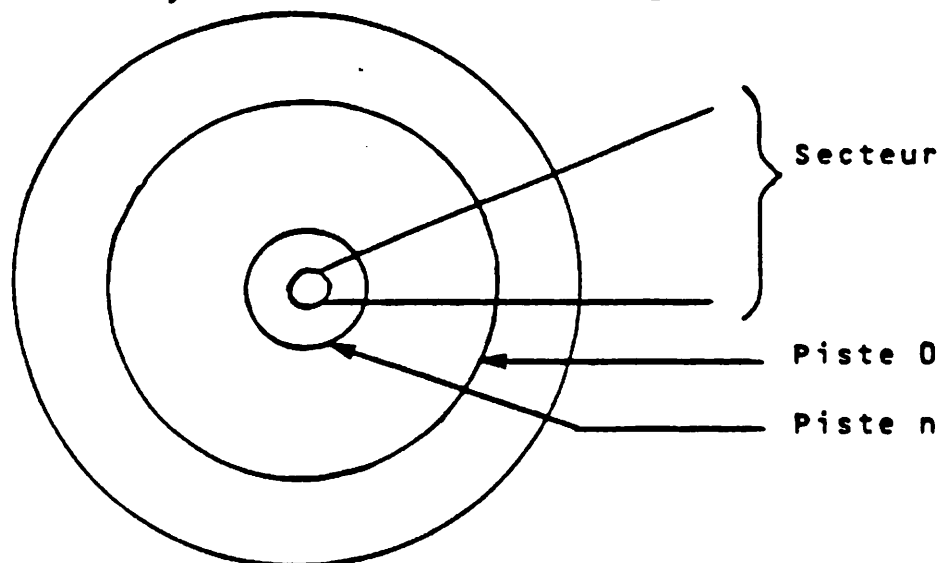


Figure 7-1. Structure des Pistes et Secteurs d'un Disque

Logiquement, le disque apparaît comme une chaîne continue de secteurs allant de la piste 0, secteur 0, à la piste 0, secteur n, puis la piste 1, secteur 0, etc. jusqu'à la piste n, secteur n.

7.3.1 Le Granule - L'Unité de Stockage Logique

La plus petite unité de stockage logique dans le système PROLOGUE est le granule qui est formé de 2 à 256 secteurs. Le nombre de secteurs par granule est spécifié uniformément pour chaque volume par le programmeur à la création du volume.

L'espace disque assigné à chaque programme est le plus petit nombre de granules qui permet de loger le programme. Même si un programme n'a que quelques octets de longueur, il doit recevoir un granule complet. Le programmeur peut ainsi donner aux unités logiques de mémoire la taille la mieux adaptée à ses besoins. En définissant des granules de petite taille, on peut optimiser l'espace requis par un ensemble de programmes courts. Cependant, le système prend plus de temps pour chercher et trouver un fichier particulier. En définissant des fichiers avec un petit nombre de granules, on diminue le temps nécessaire pour localiser un fichier, mais on peut augmenter le gaspillage d'espace.

7.3.2 Table d'Allocation de Mémoire Disque

L'allocation de l'espace d'un disque ou d'une disquette est contrôlée par une table implantée dans le secteur 0, piste 0, dans laquelle chaque bit des 254 caractères de 8 bits représente un granule du volume. Cette table, générée simultanément avec la description de chaque fichier, permet au système de gestion des fichiers de suivre et de contrôler l'évolution de chaque fichier sur le volume. Voir le Guide du Programmeur pour le Système d'Exploitation et le Système de Gestion des Fichiers PROLOGUE pour plus de détails.

7.3.3 Considérations Importantes Concernant l'Espace

1. Si le nombre de secteurs par granule n'est pas spécifié à la création du volume, 16 est pris par défaut.
2. Compte tenu de la taille de la table d'allocation des granules - 254 caractères x 8 bits - un disque dur de 10 mégaoctets doit avoir des granules d'au moins 32 secteurs si l'on veut utiliser pleinement tout l'espace disque disponible.
3. Il est possible de copier des fichiers à partir d'un disque avec une taille de granule vers un disque avec une autre taille de granule. Il peut en résulter un certain gaspillage d'espace lorsqu'on utilise le programme de copie (CP).

Exemple 1 :

Volume d'origine
de 4 secteurs
par granule

Petit programme utilisant un seul granule et un seul secteur dans ce granule : 3 secteurs gaspillés.

==>

Volume de destination
de 16 secteurs
par granule

Même petit programme utilisant un seul granule (et un seul secteur) = 15 secteurs gaspillés.

Exemple 2 :

Volume d'origine
de 16 secteurs
par granule

Petit programme d'un seul secteur utilisant seulement un granule = 15 secteurs vides.

==>

Volume de destination
de 4 secteurs
par granule

Même petit programme utilisant quatre granules de quatre secteurs chacun = 15 secteurs vides.

Dans ce dernier exemple, il ne faut pas davantage d'espace sur le volume de destination, mais il n'y a pas d'espace économisé non plus, parce que CP copie le même nombre de secteurs.

Si l'on veut copier un programme source, il est possible d'économiser de l'espace dans une situation telle que celle-ci, en se servant de l'éditeur, parce que l'éditeur reconnaît le caractère fin-de-fichier, 1A, (CP ne le reconnaît pas) et copiera un fichier source dans le plus petit nombre de granules possibles.

7.4 PARAMETRES DU SYSTEME

Un certain nombre de paramètres sont assignés à des emplacements fixes en mémoire et sont définis à la configuration du système. Ces emplacements de mémoire et paramètres (avec éventuellement leur valeur usuelle) sont définis ci-dessous.

<u>Adresses</u>	<u>Paramètre</u>	<u>Valeur normale</u>
80H	Adresse de mémoire la plus haute disponible pour l'utilisateur (notation inverse)	Dépend du système 32 K : 807FH max 8031 : 80EAH max 8021 : 80FBH
82H	Nombre de lignes sur un écran d'affichage	idem 18H
83H	Nombre de colonnes par ligne affichée	idem 50H
84H	Nombre de lignes par page d'impression	idem 3CH : 60 lignes
85H	Nombre de colonnes par ligne d'impression	idem 84H : 132 col

Assignation des caractères de commande du clavier et de l'écran

			<u>Car. ASCII</u>	hex.
86H	Echappement (si mode Echappement utilisé)		ESC	1B
87H	Flèche à droite	→	HT	09
88H	Flèche à gauche	←	BS	08
89H	Flèche vers le haut	↑	VT	0B
8AH	Flèche vers le bas	↓	LF	Saut de Ligne 0A
8BH	VIDAGE (Ecran) Effacement		FF	Saut de Page 0C
8CH	(Restauration) curseur	↖		IC
8DH	TAB		()	5C
8EH	ANNULER LIGNE (Annulation)		(CTRL-C)	03

Définitions concernant L'Unité Support Système

93H	Nb lignes graphiques	60H	: 96
94H	nb colonnes graphiques	A0H	: 160
95H	Contrôleur logiciel de l'unité de support système par défaut (pour charger PROLOGUE)	10	= Disque Souple 12 = Disque Dur
96H	Numéro d'unité pour l'unité ci-dessus	0	= Unité 0 1 = Unité 1
97H	Contrôleur logiciel de l'unité de support utilisateur par défaut (pour le chargement de programmes utilisateur)	10	= Disque Souple 12 = Disque Dur
98H	Numéro d'unité pour l'unité ci-dessus	0	= Unité 0 1 = Unité 1

Adresse de Chargement

Valeur normale

99H	Adresse implicite de	Dépend
9AH	chargement (notation inverse)	du système
9BH	Année (2 octets)	80
9DH	Mois	01
9EH	Jour	01
9FH	Numéro de jour (2 octets)	0001

<u>Eléments divers</u>		<u>Valeur normale</u>
A1H	Heures	0
A2H	Minutes	0
A3H	Secondes	0
A4H	1/10ème de seconde	0
A5H	1/100ème de seconde	0
A8H	Octet langue : définit la langue dans laquelle sont visualisés les messages 0 - Français 1 - Anglais	0
A9H	Définit la codification de la marque décimale pour le langage BAL "." point décimal 2EH "," virgule décimale 2CH	"," 2C

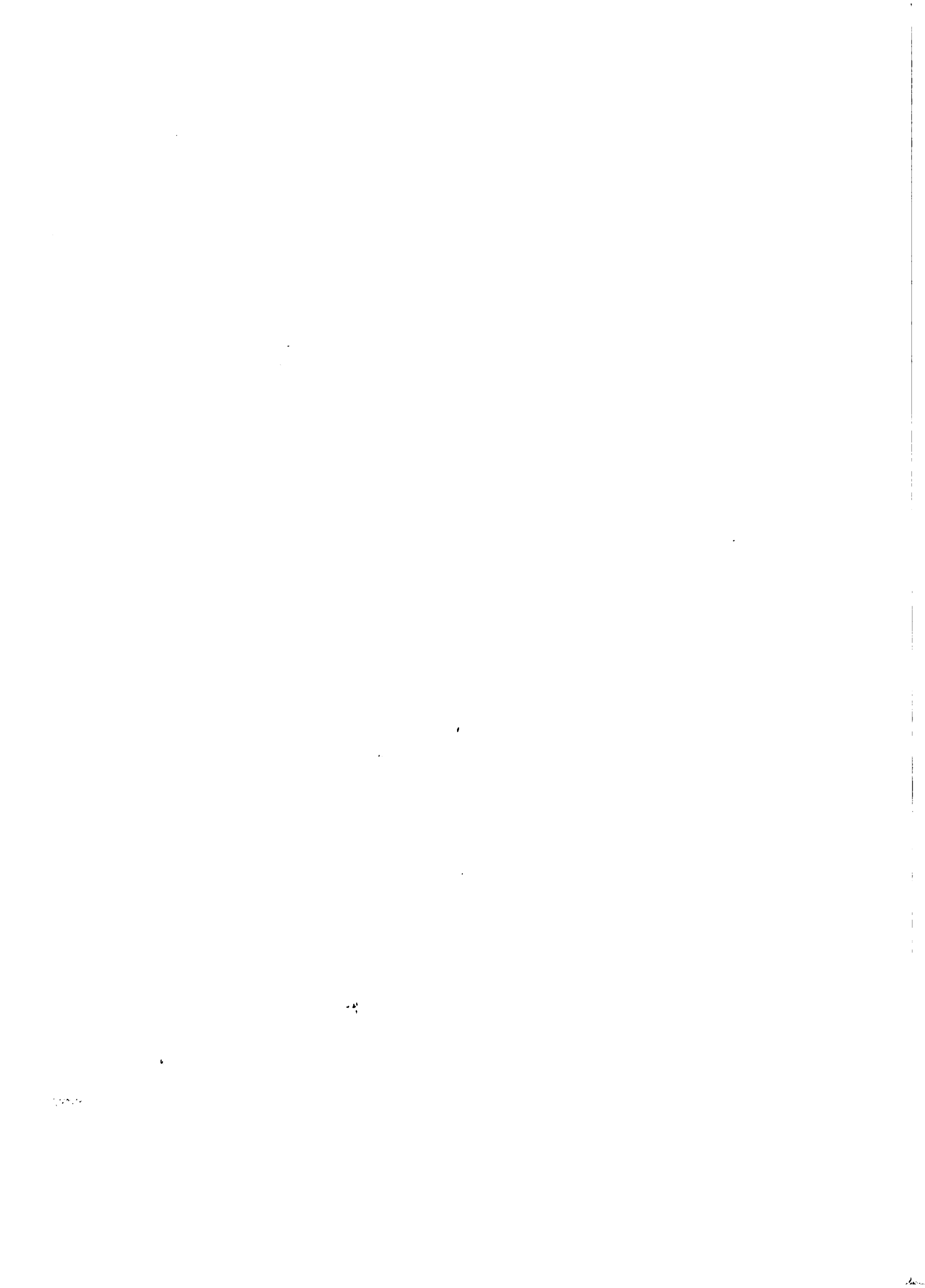
Table d'Assignment des Périphériques

6 octets sont utilisés pour décrire chaque périphérique :

2 Octets	- Nom mnémorique pour le type de périphérique
1 Octet	- Numéro d'unité
1 Octet	- Numéro de ressource
2 Octets	- Adresse d'appel (notation inverse)
COH	- Départ de table, première périphérique
C6H	- Deuxième périphérique
etc...	

Table des Noms Mnémoniques

FL	Unité disque souple de 50 pouces
DS	Disque Dur (10 ou 20 MB)
LO	Imprimante
CO	Ecran d'affichage
CI	Clavier
FS	Disque Souple de 8 pouces
BM	Bande Magnétique
MD	Minidisque 5 pouces dur



ANNEXE A. CODES D'ERREUR DE PROLOGUE

A.1 GENERALITES

Chaque fois qu'une erreur se produit dans l'un des programmes sous PROLOGUE, un message d'erreur est produit, contenant un code d'erreur. Les codes d'erreur de PROLOGUE sont listés dans cette Annexe, regroupés en Erreurs de Périphérique, Erreurs de l'Interpréteur des Commandes et Erreurs de Gestion des Fichiers.

En plus des erreurs Système PROLOGUE, des erreurs spécifiques peuvent se produire lorsqu'on travaille sous BASIC et BAL. Se référer aux manuels de référence respectifs pour des détails sur la détection et l'affichage des erreurs sous ces deux processeurs de langage.

Dans la plupart des cas, lorsqu'une erreur PROLOGUE se produit, l'opération en cours sera abandonnée, et l'on retournera au message d'appel de PROLOGUE.

Pour chacune des erreurs listées ci-dessous, on trouvera un numéro d'erreur, une description, et éventuellement la cause et l'action suggérées. Lorsque la mesure à prendre est évidente, comme par exemple, vérifier la cause affichée et la corriger, elle n'est pas indiquée.

A.2 CODES D'ERREUR DE PERIPHERIQUES

Les codes d'erreur suivants seront sortis à l'écran lorsque le système détecte une erreur mettant en cause une unité périphérique. Ces erreurs sont indépendantes du langage utilisé, et impliquent soit le matériel, soit les routines du contrôleur logiciel.

Code d'Erreur (décimal)	Description	Cause Possible/Mesure Suggérée
01	Unité non-prête	<ol style="list-style-type: none">1. Porte de l'unité non fermée convenablement.2. Disque souple incorrectement inséré (sens dessus-dessous ou devant-derrrière)3. Unité non connectée électriquement.4. Câble signal non connecté.5. Bande de pilotage disque souple cassée ou déplacée.6. Disque dur n'a pas atteint sa vitesse opérationnelle.
02	Erreur de lecture/écriture	Généralement, c'est une erreur de lecture-après-écriture lorsque le CRC (Caractère de Contrôle par Redondance Cyclique) est incorrect. Le système a réessayé l'opération 10 fois sans réussir. Peut être un problème dû au support.

Code d'Erreur (décimal)	Description	Cause Possible/Mesure
03	1. Disque souple- Erreur de positionnement de piste	<p>1. Essayer une disquette ou une cartouche de sauvegarde.</p> <p>2. Charger immédiatement le programme de statut (voir 2.7) pour déterminer la piste et le secteur où le problème se produit. Utiliser le programme CPS (2.6) pour réécrire ce <u>seul</u> secteur à partir du disque de sauvegarde.</p> <p>Si cela ne résoud pas le problème, il y a peut-être une erreur permanente irrécupérable sur le media. Laisser le programme en place, ne pas l'utiliser ni l'effacer.</p>
	2. Disque Dur - Secteur invalide	<p>Indique que l'un des secteurs d'une piste est mauvais et a été remplacé par l'un des deux secteurs de remplacement. C'est juste un avertissement, aucune action n'est nécessaire.</p>
04	1. Secteur adressé hors des limites de l'unité	<p>En se servant d'une commande d'utilitaire, on a demandé une opération avec un numéro de secteur non disponible sur l'unité sélectionnée, par exemple secteur 600 demandé sur un disque souple à simple face qui ne contient que 560 secteurs. Cela peut aussi se produire lorsqu'on copie d'une unité à double face sur une unité à simple face.</p>
	2. Erreur d'appel de paramètre	<p>Paramètre spécifié incorrectement dans le bloc de contrôle lorsque le programme d'application fait appel au contrôleur de langage d'assembleur PROLOGUE. Se référer au Guide du Programmeur pour le Système d'Exploitation PROLOGUE.</p>

Code d'Erreur (décimal)	Description	Cause Possible/Mesure Suggérée
05	Support sélectionné est protégé contre l'écriture	Tentative d'écrire sur un disque souple avec l'onglet de protection d'écriture ou sur un disque dur avec un dispositif de protection d'écriture. Enlever la protection ou utiliser un support non protégé.
06	Erreur du contrôleur périphérique	Erreur se produisant dans le contrôleur logiciel (rarement avec le disque souple). recharger le système d'exploitation.
07	Erreur de synchronisation l'adresse de secteur	Le disque n'a peut-être pas été initialisé ou a peut-être été effacé par une erreur d'écriture, car le logiciel ne peut pas détecter le dessin d'initialisation qui convient. Utiliser le programme CP pour réinitialiser. Remarque : On peut aussi avoir cette erreur si l'on essaie d'utiliser un disque souple à simple face dans une unité à double face.
10	Erreur dans les secteurs de remplacement	<u>Disque dur seulement</u> - L'erreur s'est produite dans l'un des secteurs de remplacement d'une piste. Vérifier avec soin, car la piste toute entière est peut-être inutilisable.
11	Erreur CRC détectée	Semblable à l'erreur 02, mais dans ce cas, on sait qu'il s'agit d'une erreur CRC.
12	Table des secteurs invalides incorrecte	<u>Disque dur seulement</u> - Erreur détectée dans la table des secteurs invalides. Utiliser ce disque avec précaution. Le mieux c'est de ne pas s'en servir pour les programmes importants.
13	Débordement de la table des secteurs invalides	<u>Disque dur seulement</u> - Il y a plus de 64 entrées dans la table des secteurs invalides. Copier les fichiers importants puis réinitialiser les disques. S'il y a toujours un grand nombre de mauvais secteurs après leur réinitialisation, mettre le disque au rebut.

Code d'Erreur (décimal)	Description	Cause Possible/Mesure Suggérée
14	Piste invalide - trop de mauvais secteurs	<u>Disque dur seulement</u> - Piste invalide, n'a que deux secteurs de remplacement disponibles. Programmer de façon à éviter cette piste.
15	Format dans un secteur est mauvais	Format détruit dans un secteur. Isoler le secteur (utiliser le programme Status) et éviter ce secteur, ou réécrire le secteur, ou réinitialiser le disque.

A.3 CODES D'ERREURS DU MONITEUR TEMPS RÉEL (MTZ)

Code d'Erreur (décimal)	Description	Cause Possible/Mesure Suggérée
20	Fonction invalide	Demande de création d'une ressource. N# Tâche nul dans une activation. Etc.
21	Déclaration de tâche incorrecte	Il manque un LD SP, stack en tête. N# de tâche nul.
22	Tâche déjà créée	
23	Tâche inconnue	Référence à une tâche ressource inexistante.
24	Nombre trop grand de tâches	Débordement des tables du moniteur (peut être modifié).
25	Tâche active	Activation d'une tâche active. Suppression d'une tâche active.
26	Fonction interdite pour une ressource	Activation d'une ressource. Suppression d'une ressource.

A.4 CODES D'ERREUR DE L'INTERPRETEUR DE COMMANDE

PROLOGUE s'attend à recevoir les commandes de l'utilisateur dans un format précis, et analyse la ligne de commandes pour reconnaître l'opération demandée. Les erreurs suivantes peuvent se produire lorsqu'une commande est spécifiée incorrectement et ne peut pas être traitée. L'action suggérée est de vérifier la syntaxe requise et de corriger la commande.

Code d'Erreur (décimal)	Description	Cause Possible/Mesure Suggérée
30	Syntaxe incorrecte pour l'unité de support	1. Nom mal orthographié - FL0 (lettre O au lieu de 0) 2. Ponctuation incorrecte - Espace illégal inséré ou virgules manquantes ou mal placées.
31	Numéro de périphérique inconnu	Peut être FL3 spécifié dans un système où il n'y a pas de disque souple numéro 3.
32	Unité non gérée par le système.	Le système a peut-être un disque dur, mais l'opérateur cherche à utiliser une version de PROLOGUE générée pour travailler seulement avec des disques souples.
33	Type de fichier incorrect	Un type de fichier particulier est exigé pour la commande, mais le fichier spécifié est du mauvais type. Par exemple, l'Editeur de Liens n'accepte que des fichiers de type -0 (objet).
34	Syntaxe d'adresse de chargement incorrecte	1. Adresse de chargement non précédée du "et commercial" (&). 2. Adresse spécifiée non hexadécimale.

Code d'Erreur (décimal)	Description	Cause Possible/Mesure
35	Référence externe non résolue dans fichier objet	Tentative de charger un fichier objet qui contient une référence externe non résolue à un autre fichier objet. Voir les descriptions de l'Assembleur et du Chargeur-Editeur de Liens dans le Guide de l'Opérateur pour le Système d'Exploitation PROLOGUE.
36	Fichier objet enregistré incorrectement	Fichier objet non dans le format standard RZE. Voir description de l'Editeur de Liens ou de l'Assembleur. Réassembler ou recopier.
38	Tentative de charger au-delà de la fin de mémoire	Programme trop grand pour la mémoire d'utilisateur disponible. Réduire la taille ou (si possible) charger plus bas dans la mémoire.

A.5 ERREURS DE GESTION DE FICHIERS

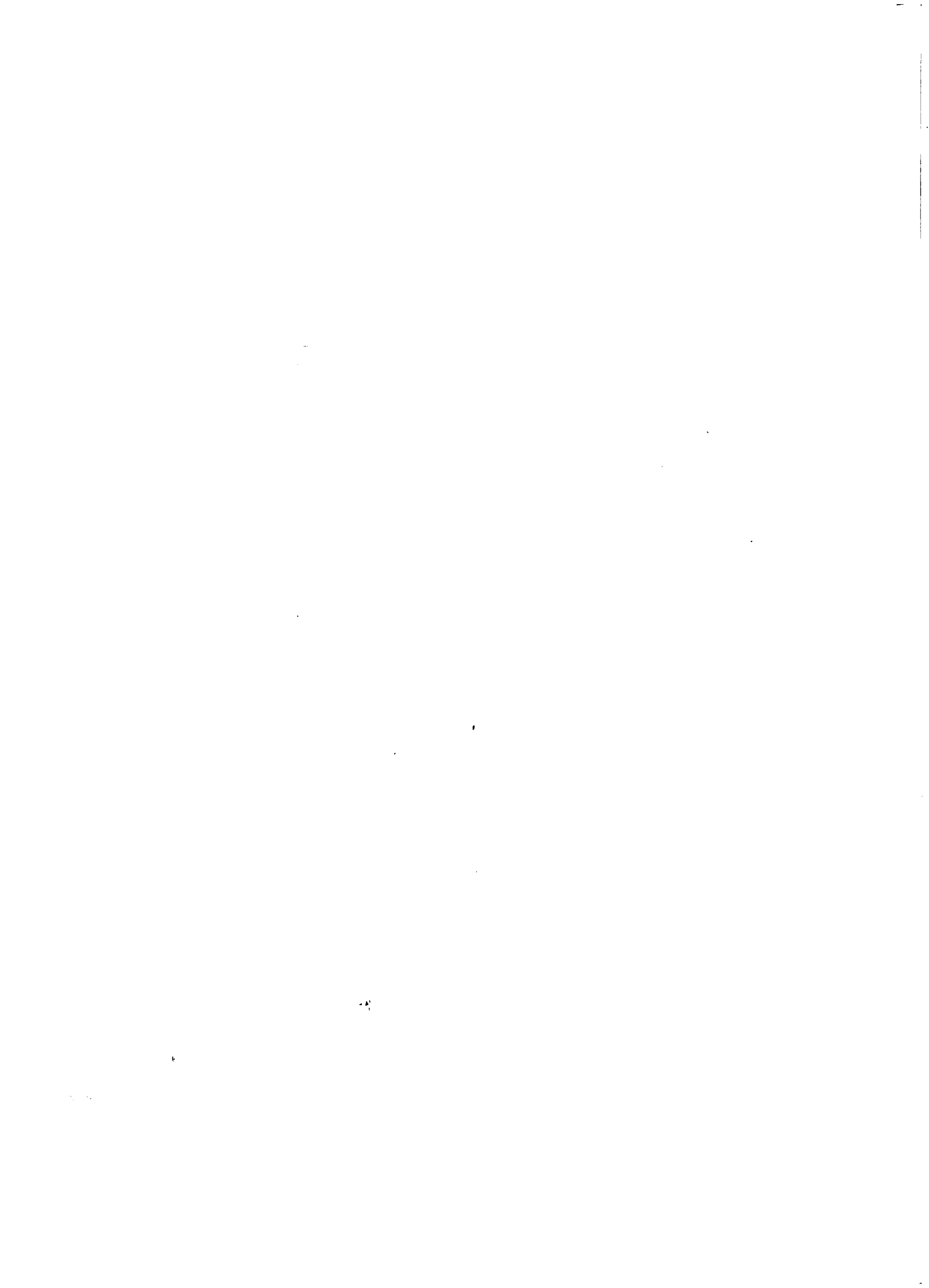
Les erreurs suivantes peuvent être détectées pendant la gestion et la manipulation des fichiers par le système PROLOGUE.

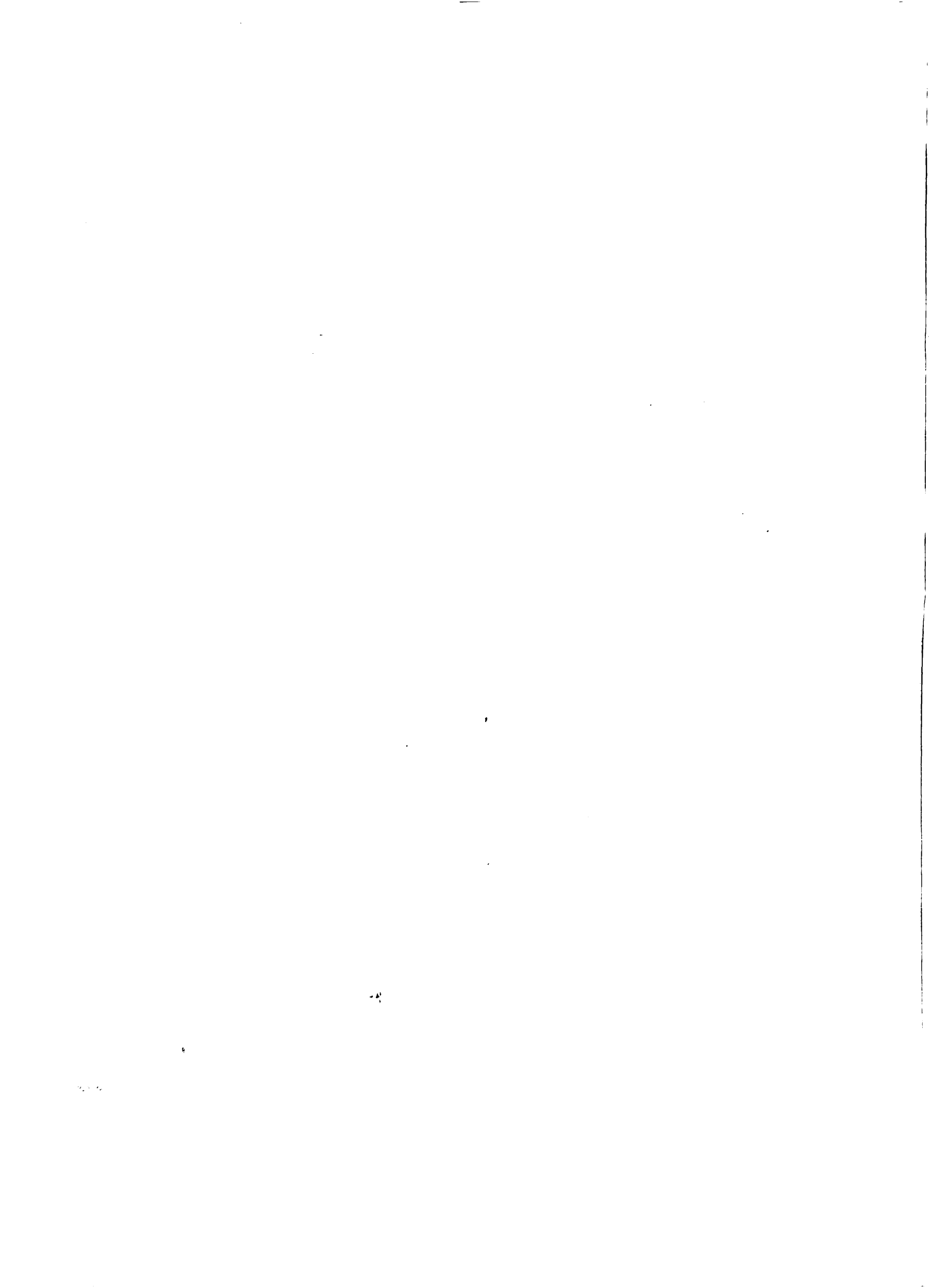
Code d'Erreur (décimal)	Description	Cause Possible/Mesure Suggérée
40	Fichier non trouvé	<ol style="list-style-type: none"> 1. Le fichier n'est peut-être pas sur le disque. 2. Il est peut-être nécessaire de spécifier le type de fichier. Par exemple, la commande de Suppression de Fichier exige le type ; beaucoup d'autres commandes non. 3. La commande spécifique peut-être une mauvaise unité.
41	Fichier existe déjà	Un fichier avec le nom de fichier spécifié existe déjà sur l'unité spécifiée.
42	Fichier fermé	En utilisant les appels en langage assembleur aux contrôleurs de PROLOGUE, on a tenté de lire un fichier sans l'avoir ouvert.

ANNEXE B. BIBLIOGRAPHIE

Se référer aux documents suivants pour des détails supplémentaires sur les sujets discutés dans ce manuel :

Titre	Publication N ^o .
Guide du Programmeur pour le Système d'Exploitation et le Système de Gestion de Fichier	R-1004
Manuel de Référence BASIC R2E	R-1005
Manuel de Référence BAL R2E	R-1006





Changement de Nom de
Fichier : [Unité.]CP,RF,Nomfichier1,Nomfichier2

Reproduction Volume : [Unité.]CP,DV,Vol1,Vol2[,RZ]

RZ - Suppression des fichiers
sur le volume cible

Changement de Nom de
Volume : [Unité.]CP,RV,Unité,NouveauNomVolume

UTILITAIRE DE COPIE DE SECTEURS (CPS)

Mode copie totale : [Unité.]CPS,Unité-S,Unité-D[,PM][,SD][,SDP][,NBPIS]

Unité-S	- Unité source
Unité-D	- Unité de destination
SD	- Simple face à double face
PM	- Prémarquage
SDP	- Simple densité de piste à double
NBPIS	- Nombre explicite de pistes à convertir.

Mode copie libre : [Unité.]CPS(cr),(cr)

Répondre aux messages d'appel qui
demandent origine, destination, piste et
secteur.

UTILITAIRE DE REPERTOIRE (/)

Répertoire de Volume [Unité.]/[Unité2][,LIS=L0][,GR]

LIS=L0	- Sortie sur imprimante
GR	- avec information sur l'implantation des fi- chiers.

Sélection de Fichiers [Unité.]/,[Unité2.]Nomfichier[,LIS=L0][,GR]

Nomfichier	- Nom de fichier spécifi- que ou générique
GR	- Avec information sur l'implantation des fi- chiers.

UTILITAIRE DATE

[Unité.]DATE(cr) ou
[Unité.]DATE,JJ/MM/AA/HH/MM

JJ/MM/AA/HH/MM - Jour/Mois/Année/Heure/
Minute

UTILITAIRE EDITEUR (ED)

Appel : [Unité.]ED,Nomfichier1[,{CF
DEST=Nomfichier2 }]

CF	- Création de fichier
DEST=Nomfichier2	- Nom du fichier de destination
Commandes de Mises à jour :	
Espace	- Avancement à l'instruction suivante
A	- Avancement à la fin de fichier
I	- Insertion de texte
R	- Remplacement et insertion
J:Nomfichier	- Insertion de bloc de code source
M	- Modification
C	- Complément
Dn	- Suppression de ligne
,Chafne	- Recherche de la chaîne dans le texte
.Chafne	- Recherche de la chaîne au début d'une ligne
*	- Répétition de la commande de chaîne
D,Chafne	- Suppression de la chaîne dans le texte
D.Chafne	- Suppression de la chaîne au début d'une ligne
Z	- Fin-de-fichier
E	- Fin de mise-à-jour
S	- Abandonner la mise à jour
L:nn	- Définition de la longueur d'une ligne de texte.

UTILITAIRE PATCH

Spécifiant [Unité.]PATCH,Nomfichier[,NSEC=no]
Nom de Fichier

NSEC=no - Secteur relatif dans le fichier

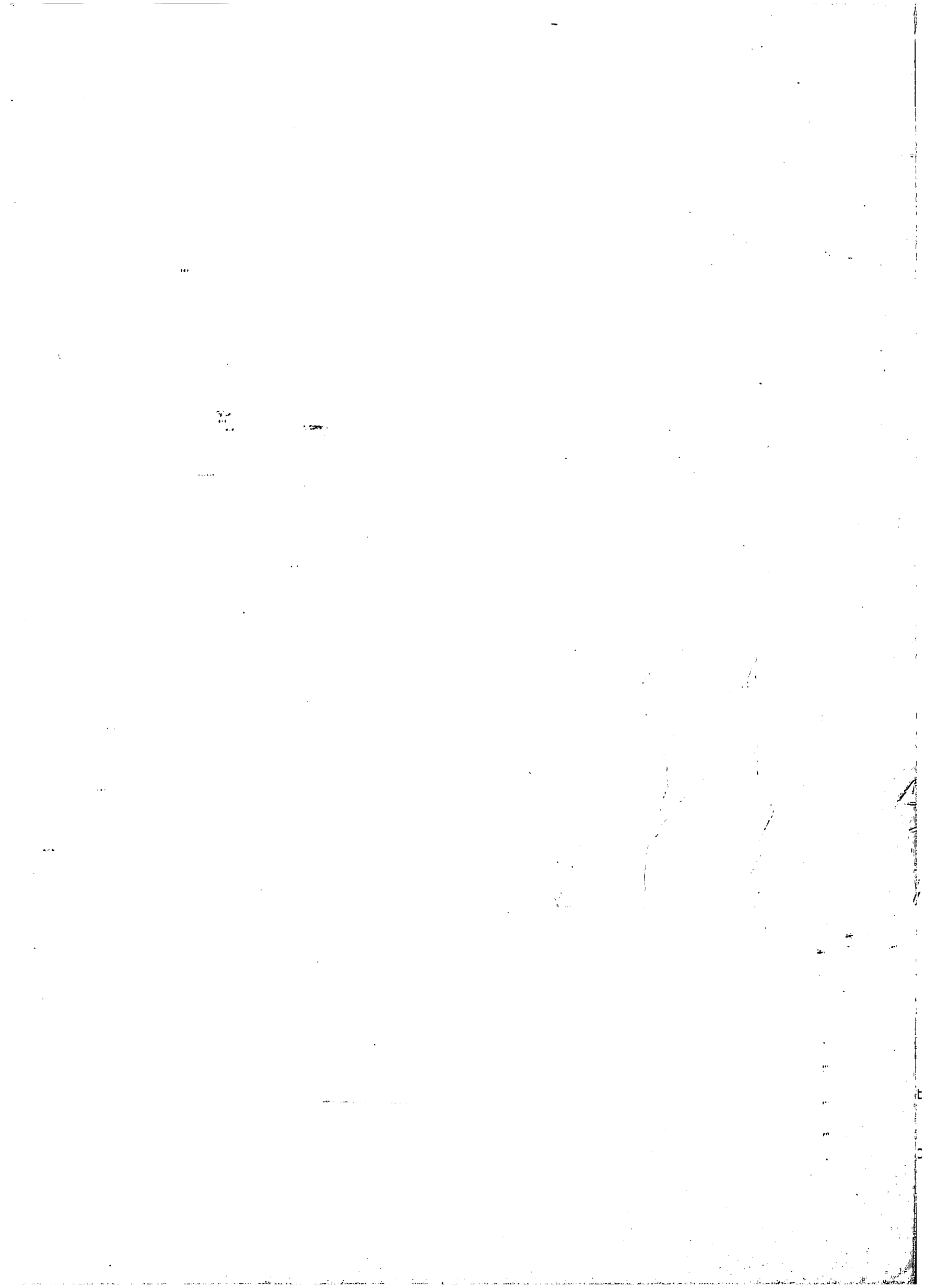
Spécifiant [Unité.]PATCH[,Unité,NPIS=Piste,NSEC=Secteur]
Piste/Secteur

Unité - Unité spécifiée
NPIS=Piste - Piste de départ
NSEC=Secteur - Secteur de départ

UTILITAIRE STATUS

[Unité.]STATUS[,Unité,LIS=L0]

Unité - Unité à vérifier
LIS=L0 - Sortie sur imprimante



XOR-423-02568
SRS1008A-998
PROLOGUE 1.9/GBASIC 5.22
Copyright (c) 1983
BULL Micral / MICROSOFT

MICRAL 8022 G - 140K

SYSTEME D'EXPLOITATION PROLOGUE DE RZE
MANUEL DE L'UTILISATEUR

Publication No. R-1003F

Edition Février 1983

